LA-UR-18-24230

Approved for public release; distribution is unlimited.

Title:	Introduction to the MCNP6 Unstructured Mesh Geometry Capability
Author(s):	Spencer, Joshua Bradly Martz, Roger Lee Alwin, Jennifer Louise
Intended for:	Attila4MC Class
Issued:	2018-05-15

Introduction to the MCNP6 Unstructured Mesh Geometry Capability

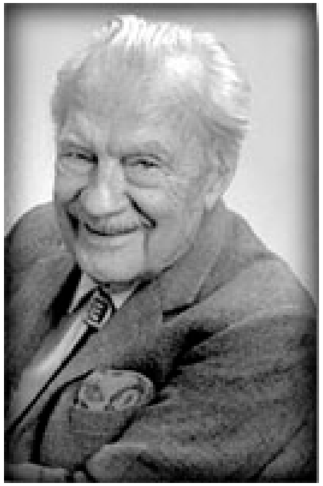
Joshua B. Spencer, Roger L. Martz, Jennifer L. Alwin

XCP-3: Monte Carlo Codes, Methods and Applications

MCNP® Trademark

MCNP® and Monte Carlo N-Particle® are registered trademarks owned by Los Alamos National Security, LLC, manager and operator of Los Alamos National Laboratory. Any third party use of such registered marks should be properly attributed to Los Alamos National Security, LLC, including the use of the ® designation as appropriate.

- **Please note that trademarks are adjectives and should not be pluralized or used as a noun or a verb in any context for any reason.**
- **Any questions regarding licensing, proper use, and/or proper attribution of Los Alamos National Security, LLC marks should be directed to trademarks@lanl.gov.**



Fun Fact:
The Term Monte Carlo was
Coined by **Nicholas Constantine Metropolis**
…
He Also Named:
Element 43: Technetium (from the Greek ‘τεχνητοζ’ for artificial)
Element 85: Astatine (from the Greek ‘αετατος’ for unstable)

Monte Carlo Method (MCM)

A Stochastic Problem Solving Method

MCM Overview

The Monte Carlo Method is Fundamentally A Cleaver Numerical Integration Scheme

Key Concepts:

Probability Density Function

$pdf : f(x)dx -$

probability a random number is between x & $x + dx$

normalization : $\int_a^b f(x)dx = 1$

Cumulative Density Function

$$cdf : F(x) = \int_a^x f(x)dx$$

$$F(a) = 0, F(b) = 1$$

Expectation Value

$$< g(x) >= \int_a^b g(x) f(x) dx \approx \frac{1}{N} \sum_{i=1}^{N} x_i$$

Variance (Uncertainty) Estimators

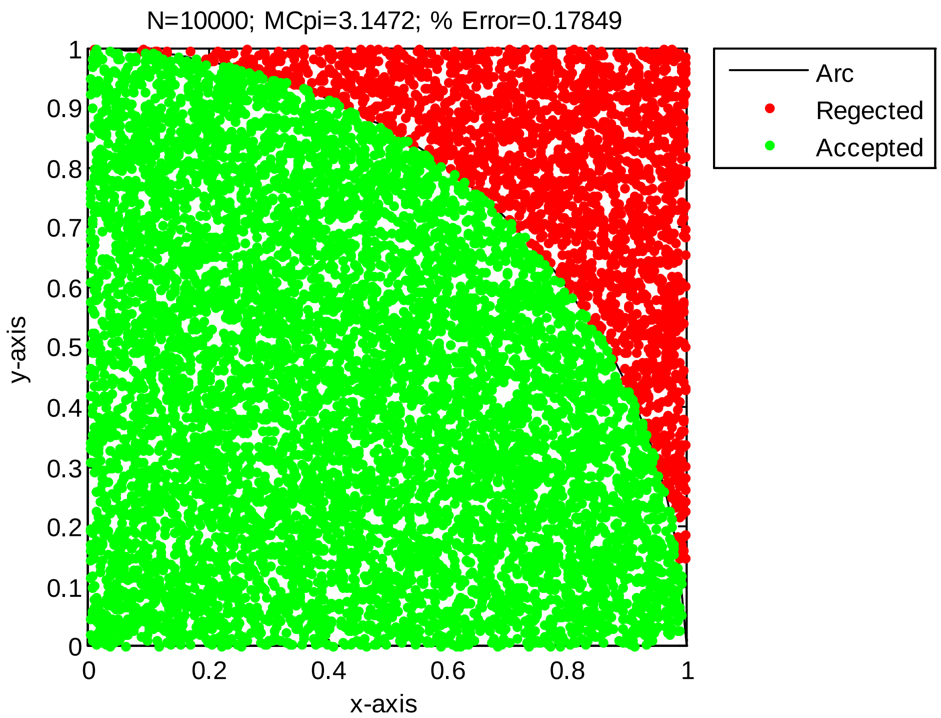
$$V(g) = \int_a^b (g(x) - < g(x) >)^2 f(x) dx$$

$$V(g) = < g^2(x) > - < g(x) >^2$$

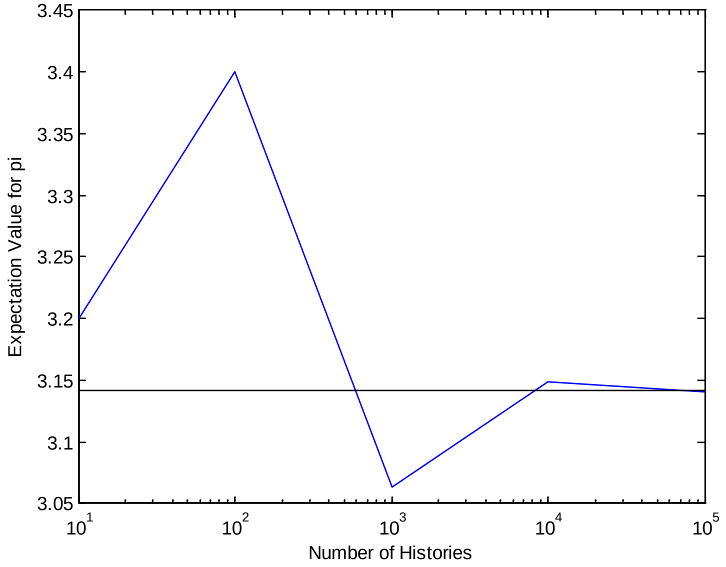
General Error Formula

$$E = \frac{\sigma}{\sqrt{N}}$$

MCM Example: Calculate π



π is proportional to the area of a circle



Convergence to π as a function of the number of histories/trials in simulation

General Error Formula $E = \frac{\sigma}{\sqrt{N}}$

Score Keeping (Tallies)

- **Score/Tallies**
  - Method to ascertain the expectation values for desired physical observables
- **Weight**
  - A number representing a particles relative contribution a tally
- **Importance**
  - The expected score generated by a particle of unit weight

Score Keeping Example:

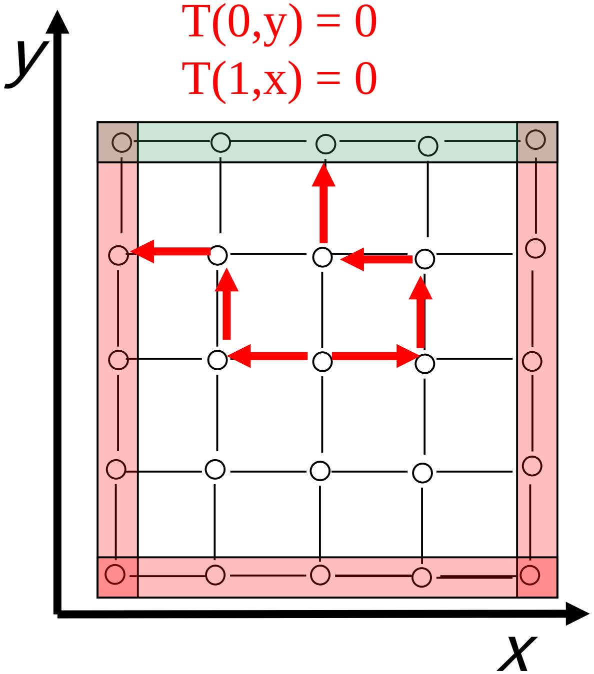
Stochastic Solution to the Laplace Equation

$T(x,1) = \sin(\pi x)$

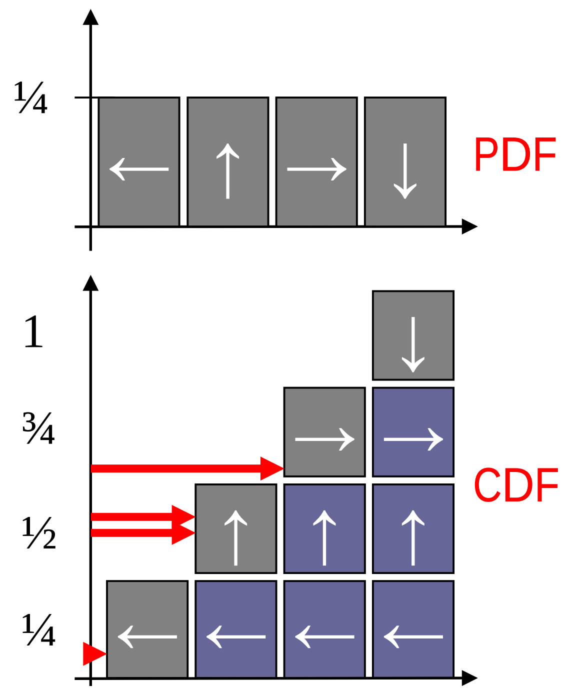
$T(x,0) = 0$

$T(0,y) = 0$

$T(1,x) = 0$



$$\nabla^2 T(x,y) = \frac{\partial^2 T(x,y)}{\partial x^2} + \frac{\partial^2 T(x,y)}{\partial y^2} = 0$$



Random Number

$$0 \le \rho \le 1$$

$\rho = 0.5570$

Score Board

1.0
0.0
0.707
+ 0.0

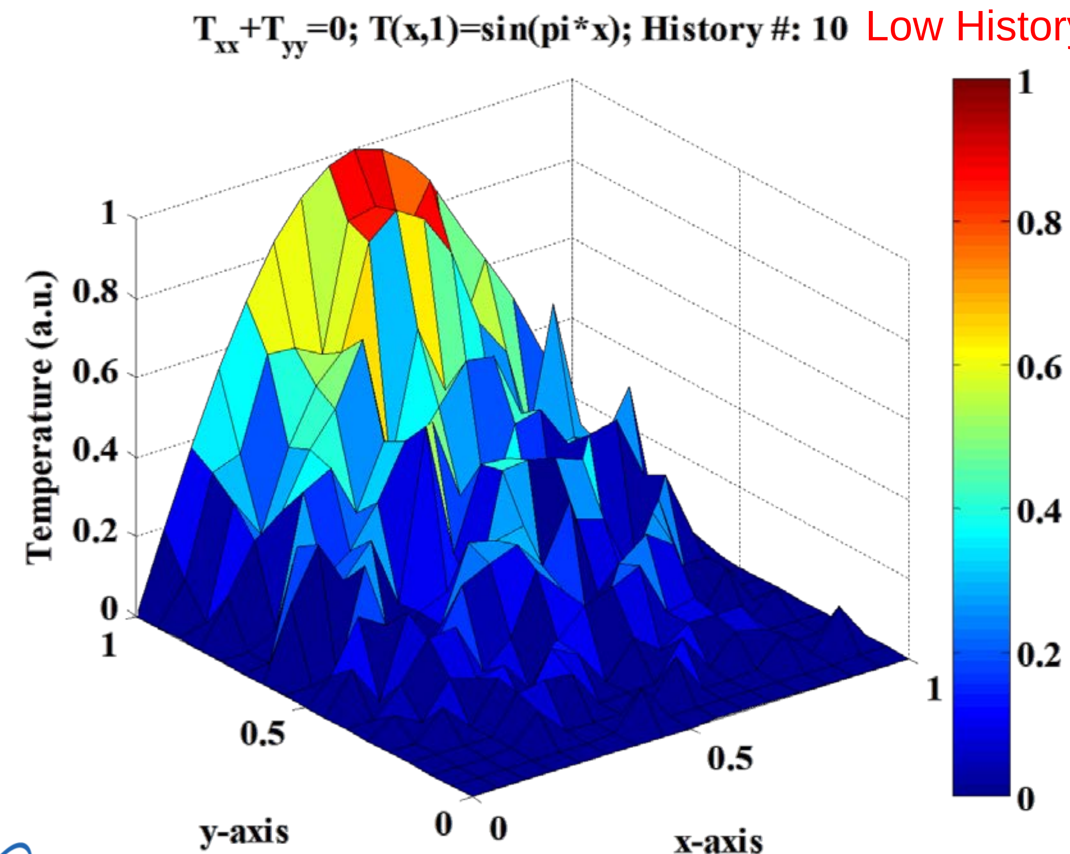
1.707 / 4

<T>=0.42675

Score Keeping Example:

Stochastic Solution to the Laplace Equation

Low History Count → High Variance Solution



Laplace Equation

$$\nabla^2 T(x,y) = \frac{\partial^2 T(x,y)}{\partial x^2} + \frac{\partial^2 T(x,y)}{\partial y^2} = 0$$

Boundary Conditions

T(x,1) = sin(π x)

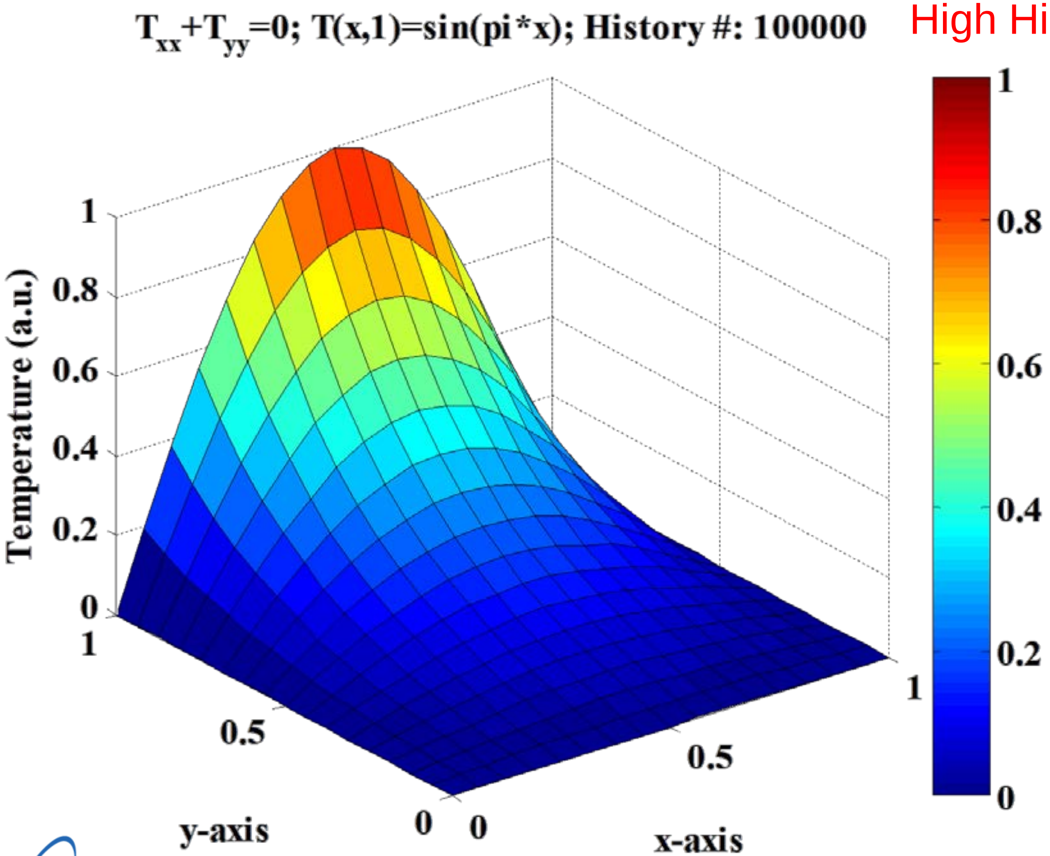
T(x,0) = 0

T(0,y) = 0

T(1,x) = 0

Score Keeping Example:

Stochastic Solution to the Laplace Equation



High History Count → Low Variance Solution

Laplace Equation

$$\nabla^2 T(x,y) = \frac{\partial^2 T(x,y)}{\partial x^2} + \frac{\partial^2 T(x,y)}{\partial y^2} = 0$$

Boundary Conditions

T(x,1) = sin(π x)

T(x,0) = 0

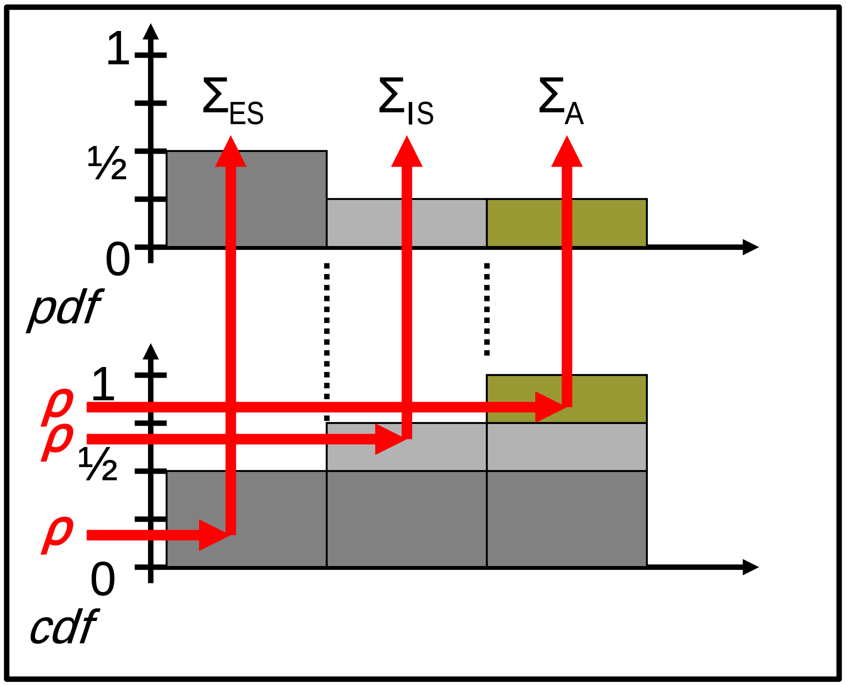
T(0,y) = 0

T(1,x) = 0

Application of MCM to Particle Transport Theory

Particle Transport is Reduced to Essentially Two Operations

1. Move the Particle (Transport Operator)
2. Collide the Particle (Collision/Interaction Operator)



Use of Macroscopic Cross-Sections Σ

$$\Sigma_t = \Sigma_a + \Sigma_{ES} + \Sigma_{IS}$$

$$pdf : 1 \equiv \boxed{\frac{\Sigma_a}{\Sigma_t}} + \frac{\Sigma_{ES}}{\Sigma_t} + \frac{\Sigma_{IS}}{\Sigma_t}$$

$$0 \leq \rho \leq 1$$

This Can Be used
To describe
Absorption, fission
(n,γ), …

Pit falls and warnings

- **For particle transport solution accuracy depends on adequately sampling phase n(x,y,z,ϑ,φ,E,t)=n(x,y,z,v_x,v_y,v_z,t)**

1		2		3		4
Change in time of neutron density	=	Change due to streaming	+	Change due to collisions	+	External sources

The Boltzmann transport equation is formulated as follows:

$$\overset{1}{\frac{\partial n}{\partial t}} = \overset{3}{\int d^3v'.\, v'\Sigma_s\left(\vec{r},\vec{v}' \rightarrow \vec{v}\right).n\left(\vec{r},\vec{v}',t\right)} \overset{2}{- \vec{v}\,.\vec{\nabla}.n} \overset{3}{- v\Sigma(\vec{r},v)n(\vec{r},\vec{v},t)} \overset{4}{+ S(\vec{r},\vec{v},t)}$$

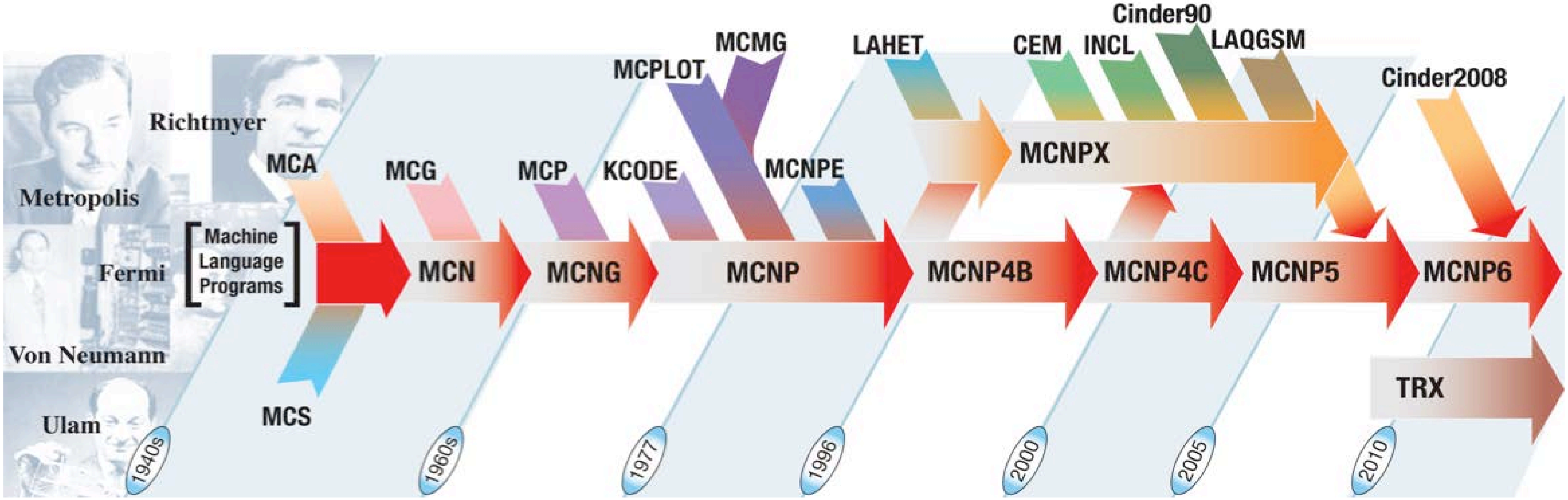
Pit falls and warnings

- **You can only simulate known physics that is included in the code**
  - In MCNP this can be cross section data or physics models
- **The MCM assumes the solution observes central limit theorem**
  - The relative error provided by MC codes is only a measure of numerical precision given included geometry, materials and physics data assumptions **not** overall solution accuracy

Historical Overview of MCNP

A Brief History

- **MCNP has existed as MCNP since the mid-1970's.**
  - Initial geometry capability was Constructive Solid Geometry (CSG)
  - Macrobody capability was added (1980's) where the macrobody surfaces were translated into CSG surfaces.
  - Current RSICC Release Version is MCNP6.2



A Brief History

- **Unstructured mesh (UM) development started in 2006**
  - Requested by LANL engineering community who standardized on Abaqus.
  - Any “code” that can write an Abaqus formatted input file with the appropriately named element sets can generate a viable mesh geometry for MCNP’s use.
- **UM has been in MCNP6 since the 1st beta release.**
  - Each subsequent release has added more UM features and fixed any associated bugs.

Abstract

The presentation is intended to introduce MCNP users to the unstructured mesh capability by discussing the basic concepts needed to construct an unstructured mesh geometry. MCNP input cards specific to this feature are shown along with examples. Limitations of the current implementation are provided along with some performance results from two simple benchmark problems.

Hybrid Geometry

Some of the discussion in the following slides will give the user a better look at what MCNP is doing behind the scenes. Not all of this is required to successfully use the Attila GUI, but nevertheless provides valuable insight.

What is the MCNP Unstructured Mesh (UM) Capability?

It is part of the new hybrid geometry capability that lets MCNP users embed "other" geometry representations of their geometry (e.g., unstructured and structured mesh) in the "legacy" constructive solid geometry (CSG) so that all representations of the geometry work together seamlessly.

What is an UM geometry?

1. **A computer aided engineering (CAE) tool such as Abaqus/CAE or CUBIT can be used to create a 3-D solid model of the entity of interest.**

   Or

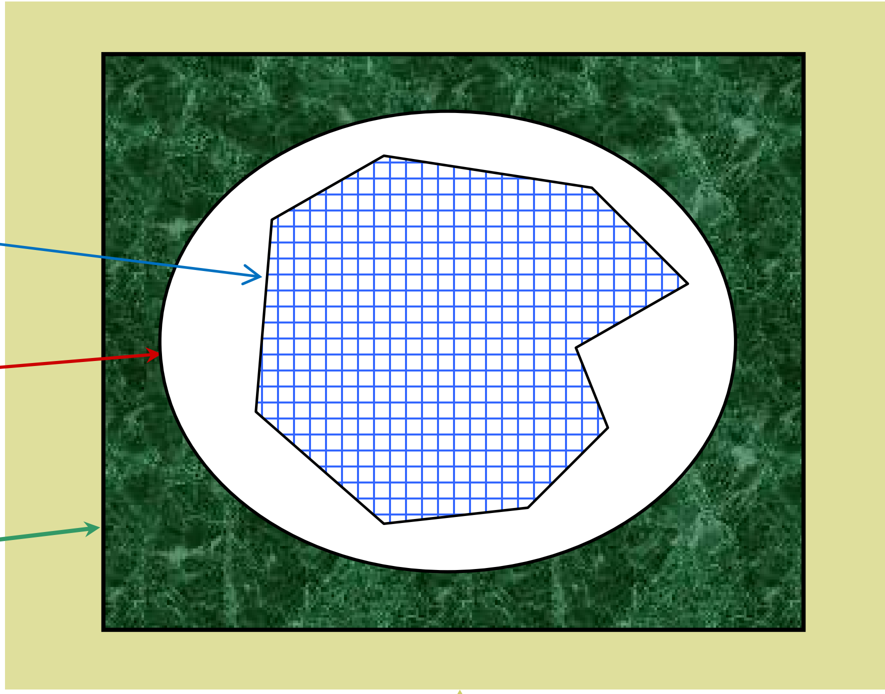
   A computer aided design (CAD) tool such as SolidWorks or SpaceClaim can create the 3-D solid model for import into the Attila4MC GUI for problem setup.

2. **An UM representation of that solid model can be created using one or all of the 6 supported finite element types:**
   - 1st order tetrahedra, pentahedra, hexahedra (Attila tetrahedra only)
   - 2nd order tetrahedra, pentahedra, hexadedra (not supported in CUBIT, Attila)

3. **The UM representation can coexist with other geometry types in MCNP in an hybrid arrangement.**

A Simple Hybrid Arrangement

Mesh universe with elliptical background cell and mesh



Unstructured or structured mesh (imp=1)

Background cell (csg filling cell; imp=1)

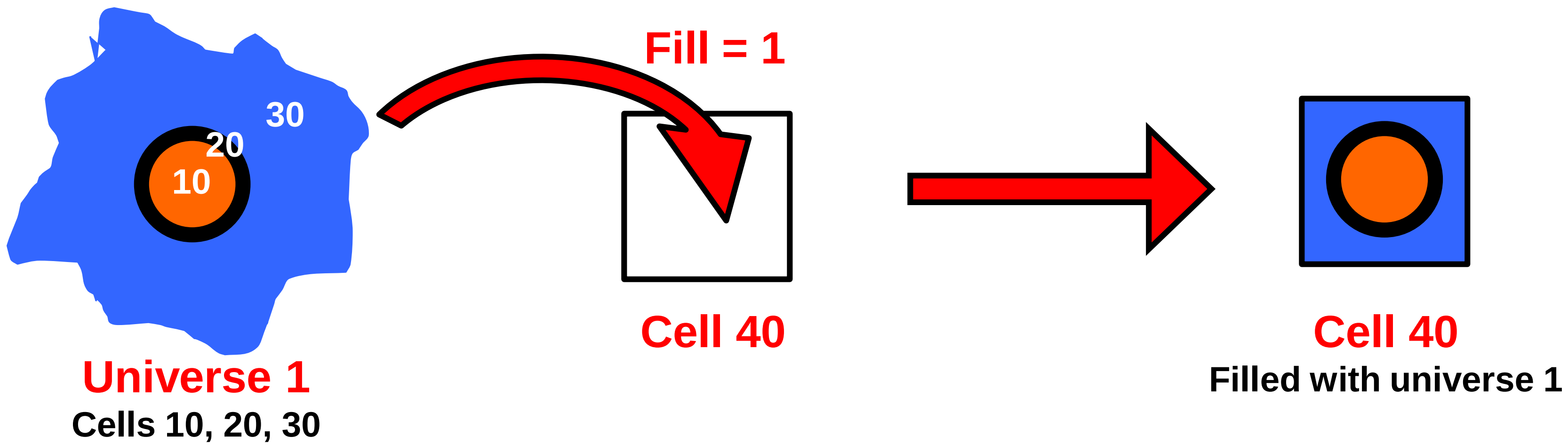
Legacy CSG cell (imp=1)

CSG outside world cell (imp=0)

Universe & Fill Concepts

- **Universe**
  - A collection of cells
  - Include a cell in universe n: put u=n on cell card, after surface list
    - n can be any number, need not be sequential
    - n must also appear on a fill= entry on another cell card (container)
    - All cells with the same u=n form a universe that fills another cell.
- **Fill**
  - Fill a container cell with a universe; insert universe into cell
  - Add fill=n on cell card, after surface list
    - n is the number of a universe
    - Usually, the cell being filled will contain a void material
    - Filled cell is a "window"
      - Clips away any part of the filling CSG universe which extends beyond the cell boundary
      - Does not clip the UM universe; user must ensure that fill cell is large enough

Universe & Fill Example w/ CSG



The MCNP6 Input

```
Problem assembly.txt   -   Fuel pin universe / fill example
c
c     CELLS
 10     100   0.06925613  -1       u=1              imp:n=1     $ fuel
 20     200   0.042910     1 -2    u=1              imp:n=1     $ clad
 30     300   0.100059     2       u=1              imp:n=1     $ water
 40     0                 -3     fill=1(1) u=2 imp:n=1     $ container
```

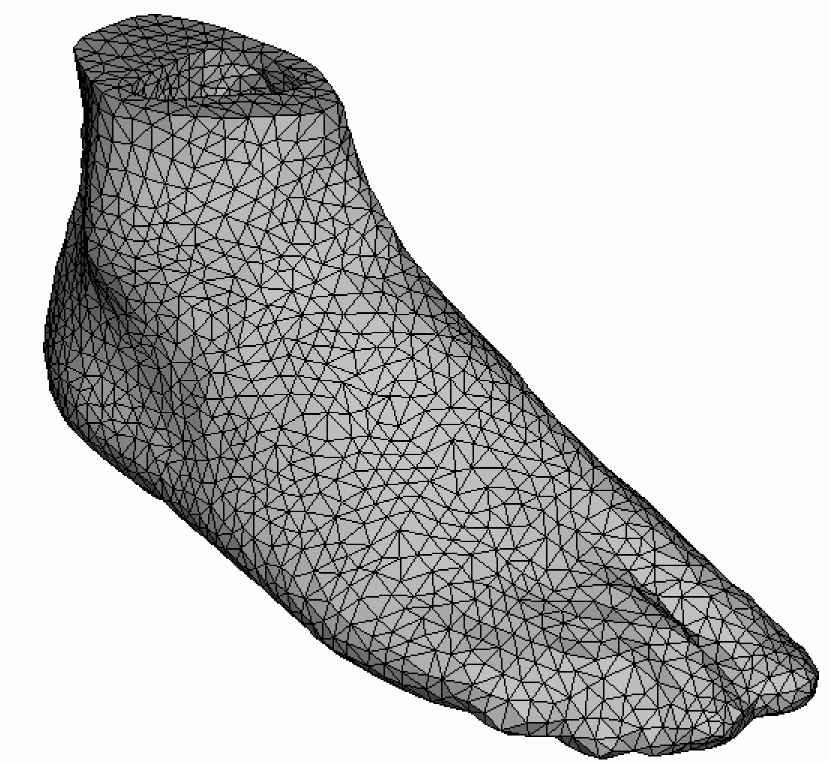
Universe Nesting Possible

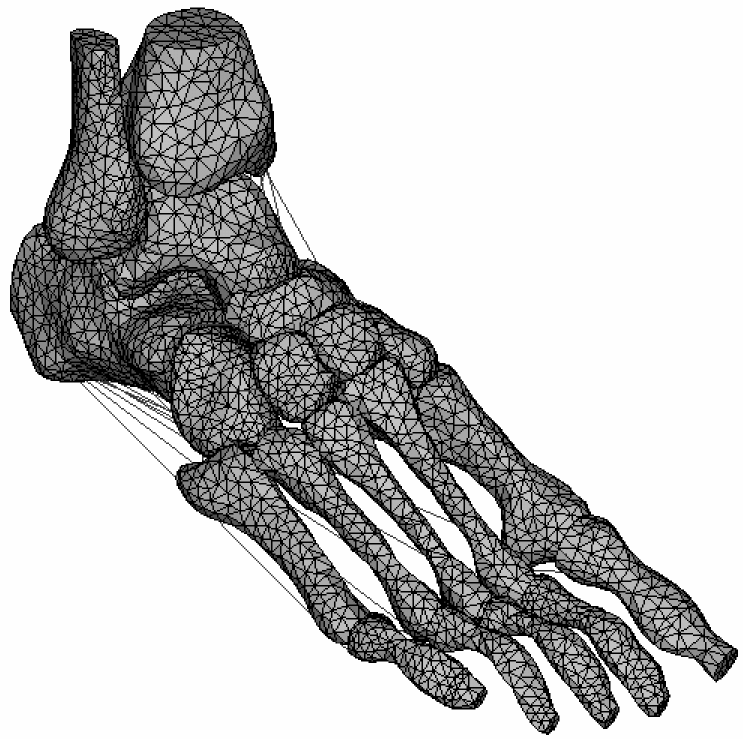
```
c     SURFACES
 1     cz    0.44
 2     cz    0.49
 3     RPP     -.7    .7     -.7    .7    0.  0

C     DATA
TR1 0. 0. 0.  0. 0. 0.  0. 0. 0. 0. 0. 0. $ Optional Translation Card
```

$\underbrace{\texttt{0. 0. 0.}}_{\text{Translation}}$ $\underbrace{\texttt{0. 0. 0. 0. 0. 0. 0. 0. 0.}}_{\text{Rotation Matrix}}$

Example





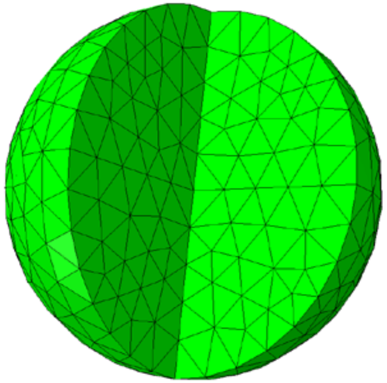
Jason Tak-Man Cheung and Ming Zhang, "Finite Element Modeling of the Human Foot and Footwear," 2006 ABAQUS Users' Conference.

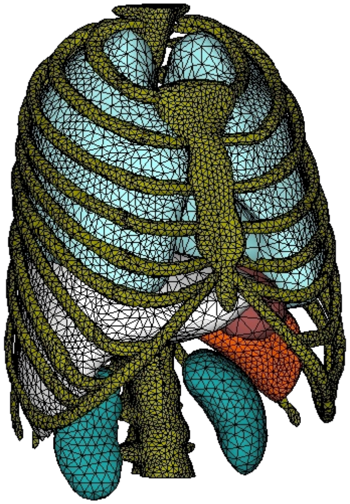
CAD vs. CAE: A Simple View

CAD – Computer Aided Design

CAE – Computer Aided Engineering

- **Both utilized solid modeling engines.**
- **Some CAD tools may generate a mesh.**
- **CAE tools generate a mesh since they are integrated with finite element methods that generally support thermo-mechanical design, analysis, and simulation.**
  - CAE is CAD on steroids





Why use UM?

1) **Easier to create complex 3-D models with a state-of-the-art CAD/CAE tool.**
2) **Better geometry and results visualization.**
3) **Easier multi-physics integration with other mesh-based physics codes.**
4) **Better calculational performance for certain problems.**
5) **It's cool!**

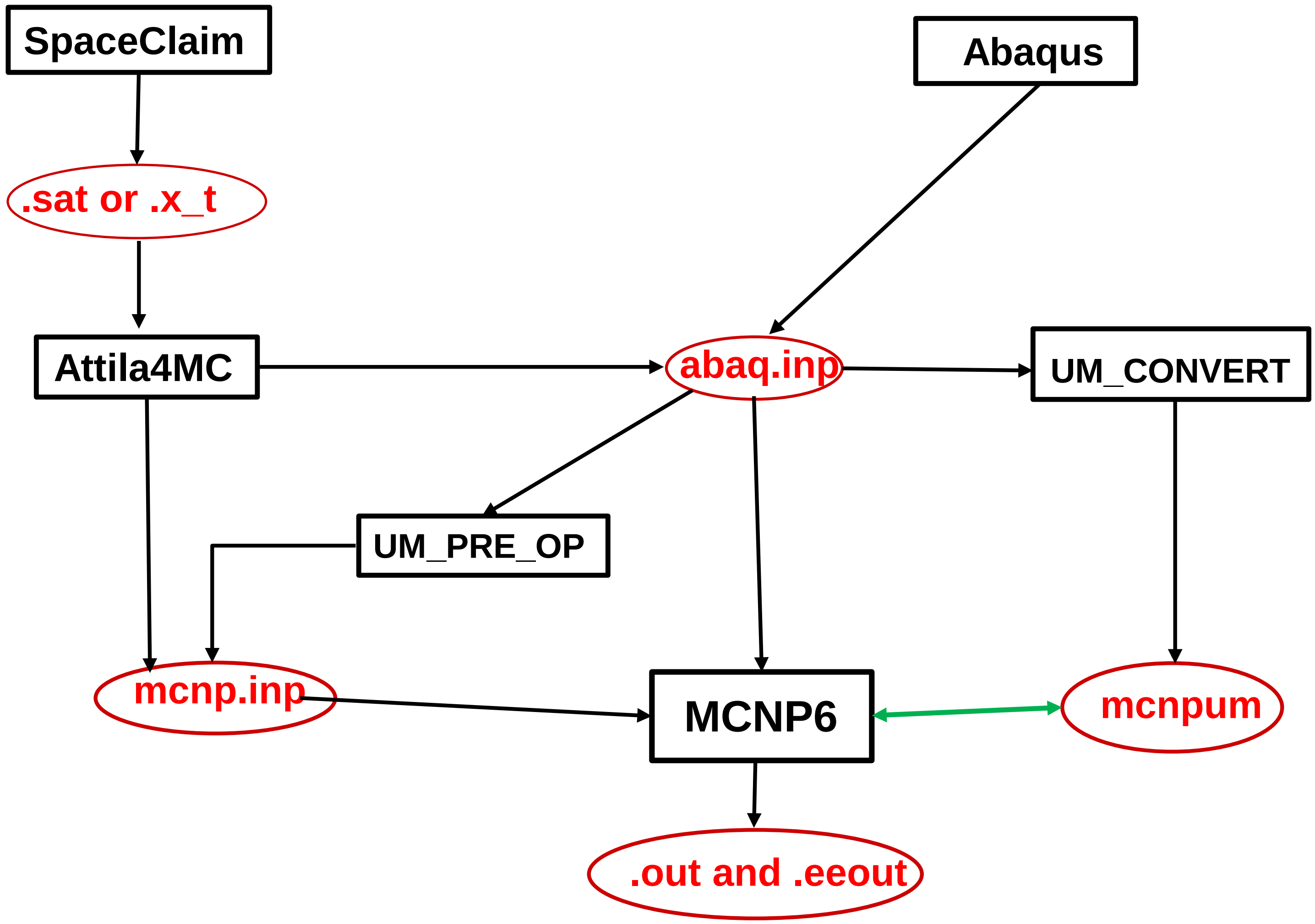

CAD 2 UM
ATTILA
Radiation Modeling Software
ABAQUS
MCNP6

Current Roads from CAD to MCNP

- **ABQAUS**
- **Primary Purpose**
  - MCNP6 transport part of larger multi-physics analysis
- **Geometry Preparation**
  - CAD geometry import capability
  - Graphical integrated geometry and mesh generation environment Supports 1st and 2nd order tetrahedral, pentahedral, and hexahedral element types
  - Python scripting interface for fine grain model development and parameter specification
- **MCNP6 Input Setup Method**
  - ***um_pre_op*** utility processes ABAQUS generated UM inp file optional user supplied data cards file to generate MCNP6 input file

- **Attila4MC**
- **Primary Purpose**
  - Radiation transport calculation set-up with hybrid Sn/MC variance reduction
- **Geometry Preparation**
  - CAD Import capability, ACIS or Parasolid format
  - Robust Mesh Generation (only supports 1st order tetrahedrons)
- **MCNP6 Input Setup Method**
  - GUI driven input setup including
    - Mesh selection and analysis
    - Materials definition and mapping with visual material verification capability
    - MCNP6 runtime parameters
    - Simple volumetric or point sources specification
    - Tally and elemental edit definition
    - Pseudo cell Importance definition
    - CADIS based weight window variance reduction

Codes & Input Files



Limitations

Limitation & Restrictions

The unstructured mesh feature is still undergoing development and has not been fully integrated with all of MCNP's existing features.

- **limited to neutrons, photons, electrons, protons and other heavy charged particles**
  - Everything but heavy, neutral particles.
  - Should not be used with magnetic fields.
- **unstructured mesh can not be placed inside a lattice**
- **a universe can not be placed within a mesh universe**
- **CSG surfaces must not clip or intersect the unstructured mesh**

Limitation & Restrictions

- the MCNP plotter supports limited shaded plots of the UM pseudo-cells by material, atom density, or mass density, mainly for positioning purposes.
- surface source reads and writes are not guaranteed to work with the unstructured mesh
- reflecting and periodic boundary conditions do not work on the mesh surfaces, but can be used with CSG surfaces
- source particles may not be started in mesh gaps
- surface tallies are not permitted in the unstructured mesh
- only pentahedra and hexahedra may appear together in a part; otherwise a part must contain only a single mesh type

Limitation & Restrictions

- **overlapping parts must not be severe; any single element may not be wholly contained within another element**
- **there are three models for use when tracking through overlapping elements; a particle tracks in an element until it finds a definite transition point in phase space (i.e., another element, gap, or background cell); a particle exits the part where it first enters the overlapping part; the boundary between the two overlapping parts is at an average position determined by the entry and exit locations along a straight flight path.**
- **even running parallel with mpi, problem setup may take considerable time if any one part has many (> 50,000) elements**

A complete list of limitations and restrictions is documented in the UM User's Guide.

Objects and Definitions

Paradigm Shift

- **Must recognize that this capability "marries" different "technologies".**
- **Each of these "technologies" was developed independently giving rise to terminology that one could consider unsettling because it is either new, different, conflicting, strange, weird, etc.**
  - Example: MCNP uses the concept of "cell" as the basic CSG building block. Abaqus uses the concept of "cell" when it refers to a 3-D region; a part may be a single cell or segmented into multiple cells.
- **One of the key concepts that was created to help the UM integrate with the MCNP world is the pseudo-cell; these are defined later. In a nutshell, this is a mechanism to let a <u>collection</u> of mesh elements have MCNP <u>cell-like</u> properties (e.g., IMP's) so that existing features are readily useable with the mesh.**

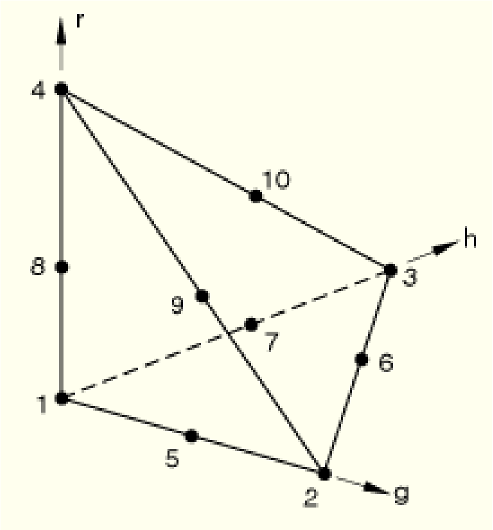
Objects and Definitions

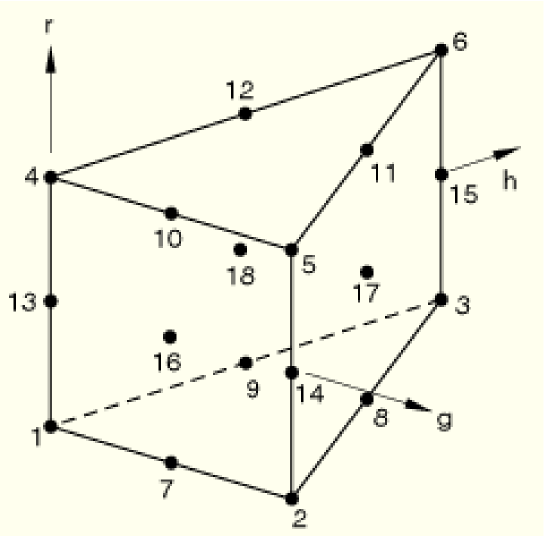
- **The following slides will attempt to define some terminology as it relates to objects associated with the unstructured mesh capability.**
- **NOTE: Some of the terminology may seem "different" from what you are accustomed. This arises due to the way the CAE / finite element users in the structural mechanics world have defined things. I have chosen to use their definitions where appropriate.**

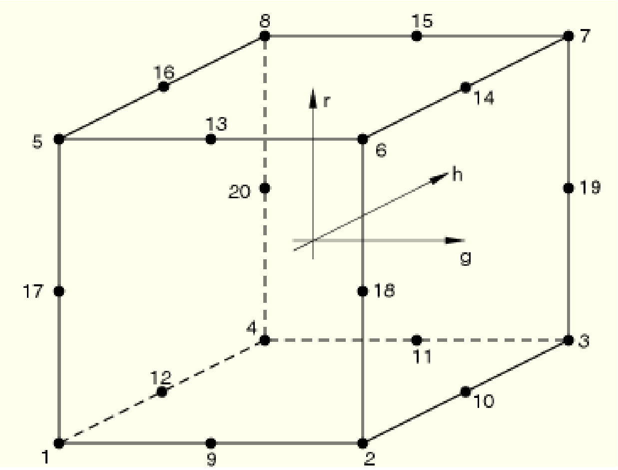
Objects and Definitions: elements or finite elements

The smallest building blocks

- **Unstructured polyhedrons with 4-, 5-, and 6-sides or faces generated by a program like the ABAQUS®/CAE. Surfaces may be bilinear or quadratic depending on the number of nodes.**







1st order – vertex nodes only; 2nd order – vertex + edge nodes

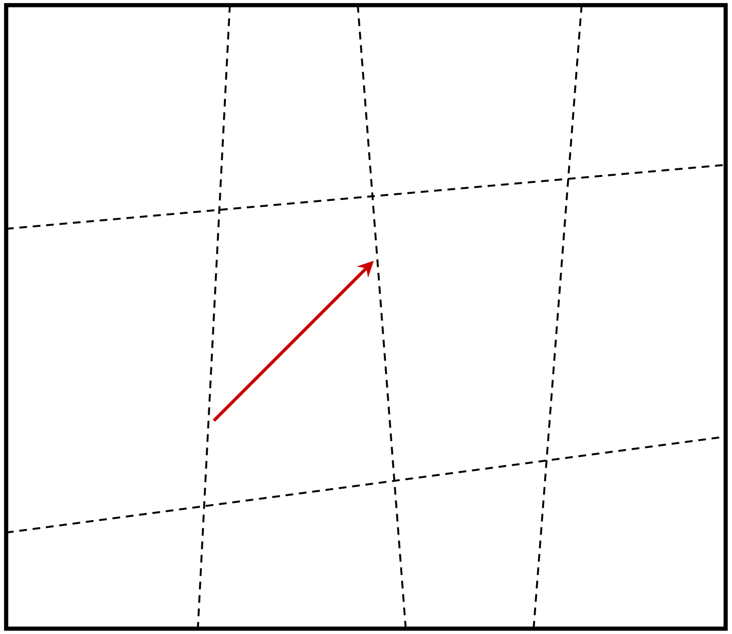
Objects and Definitions: mesh

The collection of elements comprising the entire (meshed) model.

> **NOTE: There is a distinction between the solid model and the mesh model.**
>
> **Think of the mesh as a representation of the solid model.**

Element-to-Element Tracking

- **Tracking is from face-to-face of the element (barring a collision event or any other event that MCNP checks such as distance to dxtran surface).**
- **Path length results are collected for each element through which the particle travels.**
  - Therefore, collecting results such as flux (by element) adds very little overhead to the calculation.



12-element part

Objects and Definitions: edits

- **When element-to-element tracking on the unstructured mesh is requested (this is the default), results can be accumulated on each finite element through which particles pass.**
  - This is basically free.
- **Results on the unstructured mesh are referred to as “edits” or “elemental edits” to distinguish them from the tally results in the CSG.**
  - Results go to a special output file (eeout)
- **Edits are not intended to reproduce all of the tally functionality, such as**
  - Tally types F1, F2, F5
  - Statistical analysis, statistical checks, TFC’s, empirical history score pdf, etc. (If these features are needed, use the appropriate statistical elset so that these things can be used with a collection of elements. See the discussion on statistical elsets below.)

Objects and Definitions: elsets

- **A collection of elements or a sub-set of the mesh associated with a specific tag, label, or name.**
  - Does not need to be contiguous.

Objects and Definitions: part

- **The smallest geometric object created in the CAD/CAE tool.**
  - Can contain a mesh representation
  - Can be assigned attributes such as material #'s and density
  - Can be segmented into smaller pieces; these pieces are referred to as cells; if the part is not segmented then it is also a cell

Objects and Definitions: instance

- A copy of a part used when constructing an assembly
- 1 part may be used many times

Objects and Definitions: assembly

- **The largest geometric object created in the CAD/CAE tool**
  - Constructed from instances of parts
  - A composite object
  - The final mesh model from which MCNP creates its global mesh model

Objects and Definitions: pseudo-cell

- An elset that has been mapped to an MCNP cell.
- This is a mechanism to let a collection of mesh elements have MCNP cell-like properties (e.g., IMP's) so that existing features are readily useable with the mesh.
- In essence, a group of elements (elset) has cell-like properties, but is not a traditional MCNP cell.

Pseudo: in scientific use, denoting close or deceptive resemblance to

Objects and Definitions: background cell

- **A cell that serves as the background (or container or holding) cell for the mesh.**
- **An MCNP cell into which the mesh has been placed and basically is the csg cell that is “filled” into the universe.**

Objects and Definitions: mesh universe

- **An MCNP universe comprised of the mesh and the background cell.**
  - May not contain any other lower universes

Constructing A Mesh Geometry

ABAQUS Input File Format

Parts

Assemblies

```
*Heading
** Mesh: UnitTet
** n_points = 4
** n_sides  = 6
** n_cells  = 1
** Mesh Bounding Box (cm):
**  x:    0.000 -    1.000
**  y:    0.000 -    1.000
**  z:    0.000 -    1.000
**
**  MESH
**
*Part, name=UnitTet
*Node
     1,  1.00000000e+00,  0.00000000e+00,  0.00000000e+00
     2,  0.00000000e+00,  1.00000000e+00,  0.00000000e+00
     3,  0.00000000e+00,  0.00000000e+00,  1.00000000e+00
     4,  0.00000000e+00,  0.00000000e+00,  0.00000000e+00
*Element, type=C3D4
     1,      1,      2,      3,      4
*Elset, elset=Set-material-statistic-source-UnitTet_1, generate
1, 4, 1
*End Part
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=UnitTet, part=UnitTet
*End Instance
**
*End Assembly
**
** MATERIALS
**
*Material, name=Iron_1
```

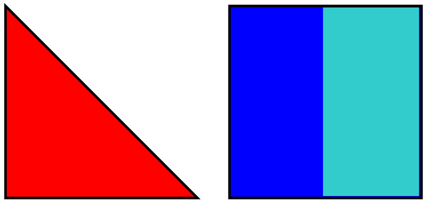
- ABAQUS Keyword
- MCNP Required
- MCNP Optional

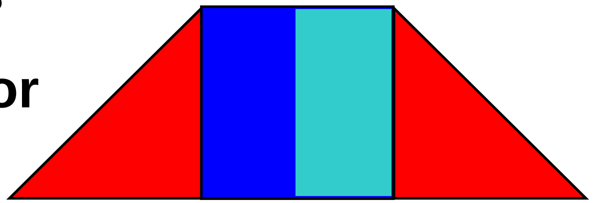
- ** ABAQUS Comment

Constructing A Mesh Geometry: Parts & Assembly

Created with CAD/CAE

- **Each "part" can consist of**
  - a single segment of one homogeneous material
  - multiple segments of different homogeneous materials (Abaqus)
- **Each "part" can be meshed independently in Abaqus/CAE**
- **The final model is an "assembly" constructed with "instances" of "parts"**
- **The mesh representation in MCNP is for the assembly (i.e., the global model)**





Constructing A Mesh Geometry: elsets

Created with ABAQUS or through the Attila GUI

- **Elements must be "tagged" with information by creating element sets (elsets).**
- **Each part must contain two elsets of data for:**
  - material sets
  - statistic or tally sets
- **Any part may contain a volume source either in its entirety or in part. If so, there must be an additional elset(s):**
  - source set(s)

Constructing A Mesh Geometry: Material Names

Created with ABAQUS

- **Material names <u>must</u> be defined so that the elset's material numbers can be matched with a meaningful name for use in the MCNP output file.**
  - Material names should contain a valid MCNP material number at the end of the name.
- **Currently, only 1 name is required and a warning is printed in the output if there are more material names expected but missing.**

Constructing A Mesh Geometry: Naming Requirements

Elsets and Material Names

Name format: ????AAAA????%ZZZ

Where AAAA is one of the keywords:

material

statistic or tally

source

Where,

% can be either the underscore character, _ , or the hyphen, - .

ZZZ is the set number and <u>must</u> be at the end of the name.

???? Are any other character or groups of characters.

Constructing A Mesh Geometry: Naming Requirements

Multi-Function Elsets

- It is possible to construct one elset that has multiple functions.
- The elset number, ZZZ, must be an appropriate number for each function. (See the following discussion.) Sets will be specified for each keyword with that number.

Name format: ???AAAA%BBBB%CCCC???%ZZZ

Where AAAA, BBBB, and CCCC are one of the keywords and there should be at least 1 that is named material:

material

statistic or tally

source

Constructing A Mesh Geometry: Materials

- **Material numbers should be unique in the UM model.**

  If part #1 is entirely lead and its material number is 1, part #2 should not use material number 1 if that part is uranium!

  If numbers aren't unique, the mesh output files (e.g., GMV file) & some utility program functions may not function properly.

- **The material names should be numbered in the same manner as the material elset numbers so that numbers are matched.**

Constructing A Mesh Geometry: Statistic / Tally Sets

- **Statistic or tally elsets are the way to group collections of elements for the purpose of volume based MCNP tallies (i.e., F4, F6, F7 tallies).**
- **Each group of statistic elsets in a part should have an unique number.**
  - With Abaqus it is possible to partition a single part into different segments. Each segment can be a different material. Segments with different materials can not belong to the same statistical set and, hence, become different pseudo-cells. This helps enforce consistency with MCNP's concept of cells.
- **The statistic elset number along with the material elset number will be used internally by the code to establish a unqiue, internal, pseudo-cell number.**

Pseudo-Cells

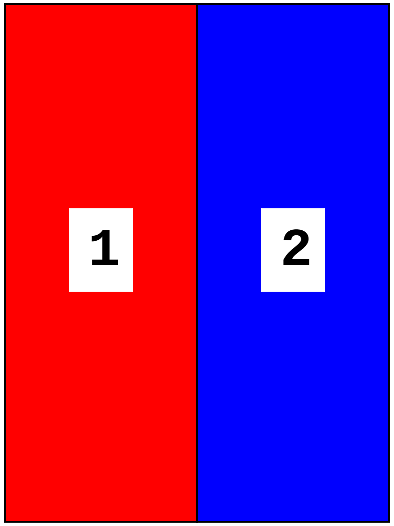
- Each homogeneous region in the mesh is referred to as a pseudo-cell.
- Want to map the mesh pseudo-cell to a legacy MCNP cell for things like cell-based tallies and variance reduction.

  Pseudo-cell could be called "mesh cell" if this terminology does not conflict with other meanings.

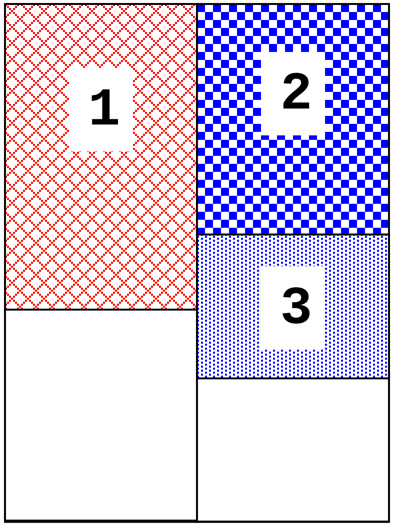
- Pseudo-cells are numbered internally starting at 1. A pseudo-cell cross reference table is printed to the output to aid the user in understanding material, tally, and part assignments between the csg and um worlds.

Constructing A Mesh Geometry

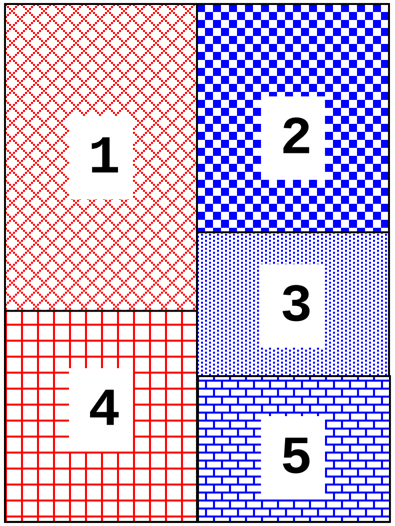
Pseudo-Cell Example



1 part with 2 materials



3 defined & 2 undefined statistical regions



5 pseudo-cells (always consecutively numbered internally from 1)

The code will tag untagged elements.

MCNP Output: Pseudo-Cell Cross Reference Table

Lists for each pseudo-cell that it knows about,

- **The corresponding MCNP cell number**
- **The instance number**
  - This is the same as the pseudo-cell number as long as there aren't any partitioned parts where partitions have different material or statistic elset numbers
- **The part number**
- **The material number**
- **The material name**

```
************************************************************************************************
* Pseudo-Cell Cross Reference Table                                                            *
************************************************************************************************
  Pseudo-Cell #    MCNP6 Cell #    Instance #    Part #    Material #    Material Name
  -------------    ------------    ----------    ------    ----------    -------------
              1              11             1         2             2    material-cube_02
              2              12             2         3            10    material-source_10
              3              13             3         1             1    material-cube__01
************************************************************************************************
```

MCNP Output: Pseudo-Cell Cross Reference Table

Example with a segmented part

```
************************************************************************************************************************
* Pseudo-Cell Cross Reference Table                                                                                    *
************************************************************************************************************************
Pseudo-Cell #    MCNP6 Cell #    Instance #    Part #    Material #    Material Name
-------------    ------------    ----------    ------    ----------    -------------
            1              11             1         2             2    material-cube_material_02
            2              12             2         1             1    material-cube_material_01
            3              13             3         3            11    material-source_material_11
            4              14             3         3            12    material-source_material_12
************************************************************************************************************************
```

MCNP Output: How The Parts Are Instanced

```
************************************************************************************************************************
*  Building the Global Tracking Model                                                                                  *
************************************************************************************************************************

Adding Instance #     1    :  part-outer_shell-1                                [Part:]   part-outer_shell

       First 1st Order TET  element number:            0      Last  1st Order TET  element number:            0
       First 1st Order PENT element number:            0      Last  1st Order PENT element number:            0
       First 1st Order HEX  element number:            1      Last  1st Order HEX  element number:           56
       First 2nd Order TET  element number:            0      Last  2nd Order TET  element number:            0
       First 2nd Order PENT element number:            0      Last  2nd Order PENT element number:            0
       First 2nd Order HEX  element number:            0      Last  2nd Order HEX  element number:            0

                                                              Last GLOBAL element         :          56
                                                              Last GLOBAL node            :         124

       Translate:      0.000000000000000     0.000000000000000     0.000000000000000
       Rotate   :      0.000000000000000     0.000000000000000     0.000000000000000
                       0.000000000000000     0.000000000000000     0.000000000000000
                       0.000000000000000

Adding Instance #     2    :  part-source-block-1                               [Part:]   part-source-block

       First 1st Order TET  element number:            0      Last  1st Order TET  element number:            0
       First 1st Order PENT element number:            0      Last  1st Order PENT element number:            0
       First 1st Order HEX  element number:           57      Last  1st Order HEX  element number:           64
       First 2nd Order TET  element number:            0      Last  2nd Order TET  element number:            0
       First 2nd Order PENT element number:            0      Last  2nd Order PENT element number:            0
       First 2nd Order HEX  element number:            0      Last  2nd Order HEX  element number:            0

                                                              Last GLOBAL element         :          64
                                                              Last GLOBAL node            :         151

       Translate:      0.000000000000000     0.000000000000000    -0.500000000000000
       Rotate   :      0.000000000000000     0.000000000000000     0.000000000000000
                       0.000000000000000     0.000000000000000     0.000000000000000
                       0.000000000000000
```

MCNP Output: How The Parts Are Instanced

```
Adding Instance #     3   :  part-middle_shell-1                              [Part:]   part-middle_shell

        First 1st Order TET  element number:            0      Last  1st Order TET  element number:            0
        First 1st Order PENT element number:            0      Last  1st Order PENT element number:            0
        First 1st Order HEX  element number:           65      Last  1st Order HEX  element number:          188
        First 2nd Order TET  element number:            0      Last  2nd Order TET  element number:            0
        First 2nd Order PENT element number:            0      Last  2nd Order PENT element number:            0
        First 2nd Order HEX  element number:            0      Last  2nd Order HEX  element number:            0

                                                               Last GLOBAL element         :          188
                                                               Last GLOBAL node            :          367

        Translate:      0.000000000000000     0.000000000000000     0.000000000000000
        Rotate   :      0.000000000000000     0.000000000000000     0.000000000000000
                        0.000000000000000     0.000000000000000     0.000000000000000
                        0.000000000000000
```

Translate & rotate using Abaqus convention

```
    Global Model Extents
    Min X: -4.00000E+00   Max X:   4.00000E+00
    Min Y: -4.00000E+00   Max Y:   4.00000E+00
    Min Z: -4.00000E+00   Max Z:   4.00000E+00
```

Global Model Extents:

- **Min & max for each mesh direction**
- **Background cell must be large enough to include these dimensions w/o clipping the mesh**
- **Doesn't reflect MCNP universe translations or rotations**

MCNP UM Input Cards

Embedded Mesh Universe

- **Geometry mesh is embedded as the lowest level universe.**
- **Cell card requirements:**
  - One cell card for each mesh “pseudo-cell”.
    - Check the mcnp “outp” file for a table showing how the code expects the “pseudo-cells” to match with the mesh regions. Correct material names in this table are dependent upon the material rules cited above.
  - There must be a cell card that serves as a “background” for the embedded mesh. Think of this as the cell that substitutes or stands-in for the mesh. Another way of stating this is that this cell is the smallest cell into which the mesh can fit. (See next slide for examples.) This can be considered the background material in which the mesh resides and is essential for the unstructured mesh to work correctly. This cell must also have a “u” descriptor. See the “background” key word on the embed card.

Embedded Mesh Universe: Cell Cards

- **Cell card requirements:**
  - There can be no cells outside of the background cell in the embedded mesh universe; put CSG cells outside the universe in order to build the remainder of the hybrid geometry.
  - The background cell and pseudo-cell descriptions appear as normal MCNP cell descriptions except they must contain a null surface; enter a zero (0) for the null surface description.

- **Example:**

```
c     *** Cell cards ***
c
      10      2  -7.8240   0              u=2   $ pseudo-cell
      11      1  -1.2230   0              u=2   $ pseudo-cell
      12      2  -7.8240   0              u=2   $ pseudo-cell
      13      3  -0.0012   0              u=2   $ pseudo-cell
      14      0            0              u=2   $ background cell
      20      0          -10     fill=2         $ fill the mesh universe
       6      0       10  -11
       7      0            11
```

MCNP Output: Pseudo-Cell Cross Reference Table

Lists for each pseudo-cell that it knows about,

- **The corresponding MCNP cell number**
- **The instance number**
  - This is the same as the pseudo-cell number as long as there aren't any partitioned parts where partitions have different material or statistic elset numbers
- **The part number**
- **The material number**
- **The material name**

```
***********************************************************************************************
* Pseudo-Cell Cross Reference Table                                                           *
***********************************************************************************************
  Pseudo-Cell #    MCNP6 Cell #    Instance #    Part #    Material #    Material Name
  -------------    ------------    ----------    ------    ----------    -------------
              1              11             1         2             2    material-cube_02
              2              12             2         3            10    material-source_10
              3              13             3         1             1    material-cube__01
***********************************************************************************************
```

Embedded Mesh Data Cards: Geometry

Embedded Mesh Control Card

```
EMBEDn  meshgeo=  mgeoin=  meeout=  meein= length=
        background=  matcell=  filetype=  gmvfile=
        overlap=
```

`n`	embedded mesh universe number n, must match a valid universe # from cell cards
`meshgeo`	mesh geometry type Current permitted values: abaqus, mcnpum, lnk3dnt
`mgeoin`	mesh geometry input file name
`meeout`	elemental edits output file name
`meein`	elemental edits input file name (valid only in continuation runs)

NOTE: all filenames must be lowercase

Embedded Mesh Data Cards: Geometry

Embedded Mesh Control Card (cont.)

```
EMBEDn  meshgeo=  mgeoin=  meeout=  meein=   length=
        background=  matcell=  filetype=  gmvfile=
        overlap=
```

`length`	conversion factor to centimeters for all mesh dimensions in input and output length parameter not used if mesh geometry was converted to mcnpum format
`background`	cell number of the background cell for the mesh
`matcell`	pairs of numbers: 1st number = mesh material number (actually, the mesh pseudo-cell) 2nd number = MCNP cell number (i.e., the pseudo-cell #)

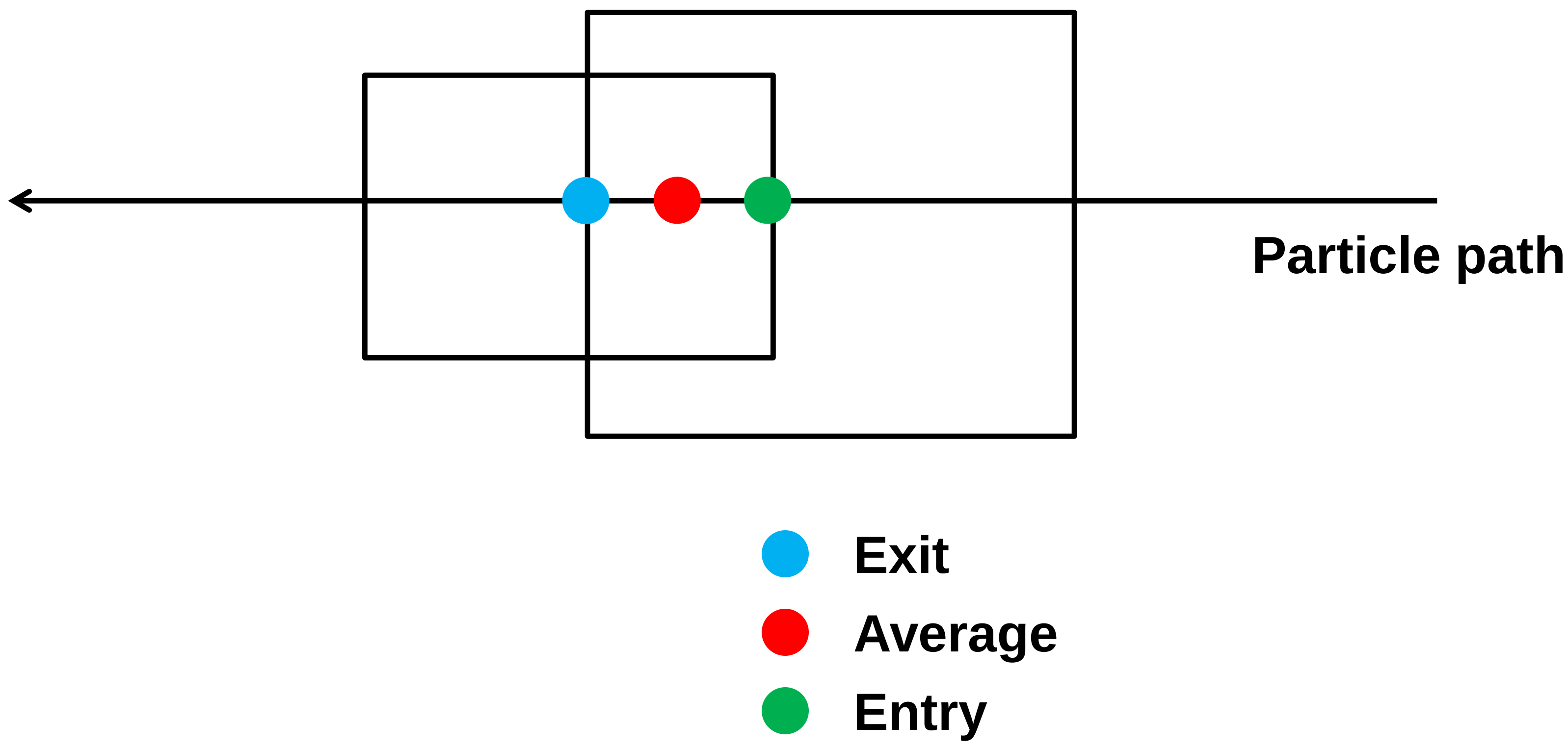
Embedded Mesh Data Cards: Geometry

Embedded Mesh Control Card (cont.)

```
EMBEDn  meshgeo=  mgeoin=  meeout=  meein=   length=
        background=  matcell=  filetype=  gmvfile=
        overlap=
```

`filetype`	output eeout file type; on input, the code determines the correct type Current permitted values: binary, ascii
`gmvfile`	output file name for the gmv geometry only file (LANL use)
`overlap`	model to treat overlapping elements Values are: entry, exit, average (see next slide)

Embedded Mesh: Overlap Models



Embedded Mesh Data Cards: Geometry

Embedded Mesh Control Card (cont.)

```
EMBEDn  meshgeo=  mgeoin=  meeout=  meein=   length=
        background=  matcell=  filetype=  gmvfile=
        overlap=
```

`overlap` 1st entry is for global use

Treatments for individual instances/parts can be specified by a second (or third) parameter followed by a list of valid pseudo-cell numbers.

Example:

```
overlap=average exit 1 entry 5 6
```

Embedded Mesh Data Cards: Edits

Elemental Edits Control Card

```
EMBEEn:<pl>  embed=  energy=  time=
```

`n`	elemental edit number ending in 4, 6, or 7; follows tally convention; indep. of tally #'s
`<pl>`	particle designator from particle list
`embed`	embedded mesh universe number; must correspond to a valid embed card #
`energy`	conversion factor from MeV/gm for all energy related output
`time`	conversion factor from shakes for all time related output
`errors`	statistical uncertainties: no (default); yes

Embedded Mesh Data Cards: Edits

Elemental Edit Energy Bins & Multipliers

EMBEBn B_1 B_2 ... B_k

n elemental edit number; 0 is not valid.

B_i monotonically increasing upper energy of the *i'th* bin.
values in units of MeV

EMBEMn M_1 M_2 ... M_k

n elemental edit number; 0 is not valid.

M_i monotonically increasing upper energy of the *i'th* bin.

Embedded Mesh Data Cards: Edits

Elemental Edit Time Bins & Multipliers

EMBTBn B_1 B_2 ... B_k

n elemental edit number; 0 is not valid.

B_i monotonically increasing upper time of the ***i'th*** bin.
values in units of shakes (1 shake = 10^{-8} s)

EMBTMn M_1 M_2 ... M_k

n elemental edit number; 0 is not valid.

M_i monotonically increasing upper time of the ***i'th*** bin.

Embedded Mesh Data Cards: Edits

Dose Energy Flux Conversion Factors

EMBDEn B_1 B_2 … B_k

Embedded elemental edit dose energy bin boundaries card

n elemental edit number; 0 is not valid.

B_i monotonically increasing upper energy of the *i'th* bin. values in units of MeV

EMBDFn M_1 M_2 … M_k

Embedded elemental edit dose function card

n elemental edit number; 0 is not valid.

B_i multiplier for the *i'th* dose energy bin. Default = 1.

Embedded Mesh Data Cards: Volume Source

Volume Source Control

SDEF **POS= ???**

??? **x, y, z position for point sampling, or volumer for unstructured mesh volume source. Note that the last character "r" stands for sampling by rejection.**

This gives uniform sampling of position over all elsets tagged with the "source" keyword. (More on next slide.)

Embedded Mesh Data Cards: Volume Source

Volume Source Sampling

- **A source elset or elsets are defined in Abaqus (see previous elset discussion).**
- **An element is selected proportional to the total source volume. That is, proportional to its volume divided by the total mesh source volume.**
- **A position is selected by rejection uniformly in the element's volume.**
  - Least efficient position sampling: tetrahedra & highly distorted elements
  - Most efficient position sampling: hexahedra

Embedded Mesh Data Cards: Multiple Volume Source

- **If the volume sources appear in different pseudo-cells and it is desired to sample non-uniformly among the pseudo-cells, use a dependent distribution where `POS` is a function of `CEL`.**
- **Only uniform sampling within a pseudo-cell is supported.**
- **Example:**

```
sdef  pos=fcel=d1  cel=d2
ds1 L volumer volumer
si2 L   101     103
sp2     0.4     0.6
```

- **Yes. You can mix and match sources.**
- **See the UM User’s Guide for more information.**

Expert Advice, Opinion, & Perspective

Using somebody's already generated CAD file

- **May contain too many irrelevant features for the MCNP model**
  - Time may be required to "de-feature" the existing model
    - SpaceClaim tools make this an easier task
- **May contain unsupported element types such as 2-D surface elements**
  - Totally un-useable without a great deal of work
- **Fidelity of the model may not be good enough for MCNP**
  - Dimensions may not be very accurate
  - Parts may not touch other parts exactly where they should
  - Time will be required to fix

Mesh & The Monte Carlo Method

The mesh in MCNP is used for 2 basic purposes:

1. **Define boundaries between parts (i.e., cells)**
   - More elements or 2nd order elements may be required to accurately represent the boundary and the associated volume and mass that it encloses.
2. **Collect edit results for visualization**
   - Mesh granularity should be increased in areas of large gradients for better representation during visualization.

Expert Advice, Opinion, & Perspective

Think Through All Requirements Before Modeling

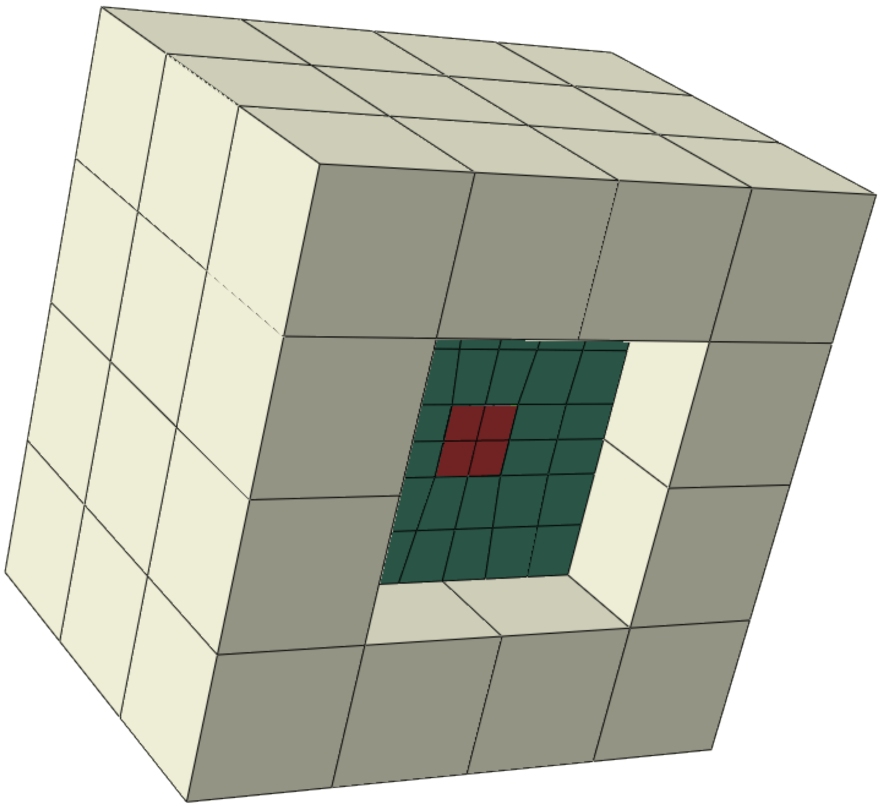
- **Fast input processing and calculation times point to parts with 50,000 elements or less.**
  - Can be accomplished with more, smaller parts or re-meshing.
- **Input processing time for models with many instances can be minimized when running with the mpi version of MCNP.**
  - Minimum number of mpi processes: 1 + number of instances in the model
- **Being able to use a certain element type may require partitioning the part.**
  - Important if hexahedra are desired.
- **Peeling away the outer layers of the model may require partitioning of the part such that removal of groups of elements result in a smooth surface.**

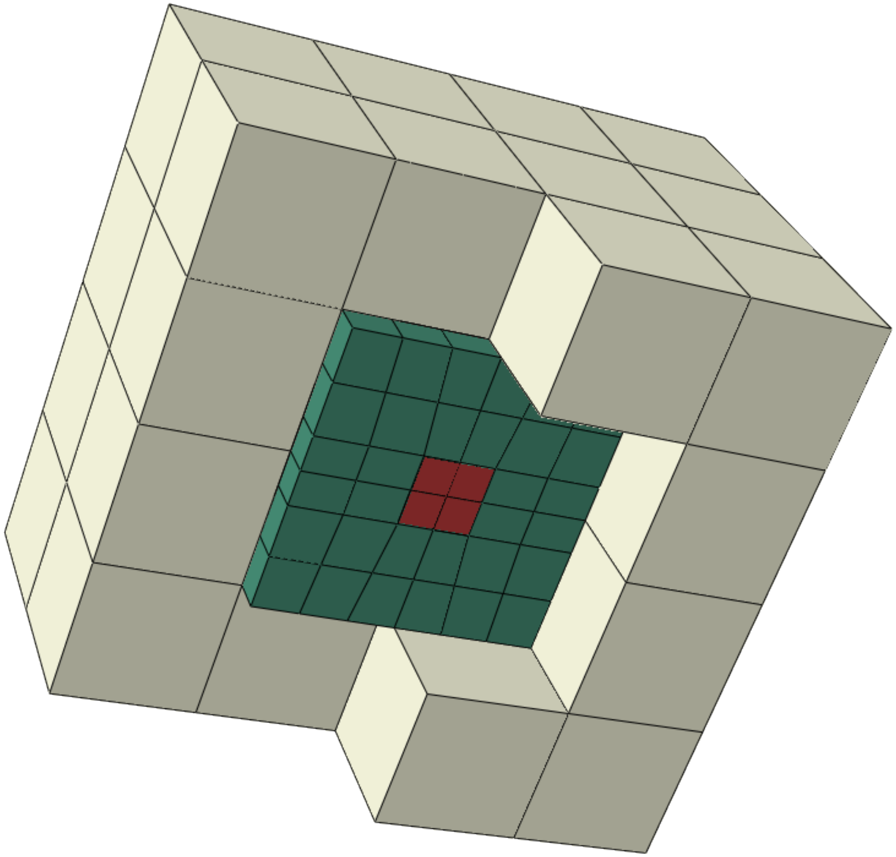
Parallel Input

- **UM problem setup requires nested-loops in several parts of the code. These loops can be time-intensive.**
- **Can speed this up by running mpi.**
  - Want 1 mpi slave node for each instance. The minimum number of mpi processes to specify should be 1 + number of instances in the model.
  - Each instance or part will then have a dedicated processor for its input. At this time, multiple processors per instance or part are not implemented and there is no load balancing.
  - Still limited by the instance / part with the largest number of elements.
- **The most efficient parallel input processing takes place when all parts have approximately the same number of elements and there is more mpi processes than instances.**
  - < ~50,000 elements per part is a key # for efficient input processing.
  - When there are fewer processes than instances – round robin.

Examples

Concentric Cubes





Concentric Cubes: MCNP Input File

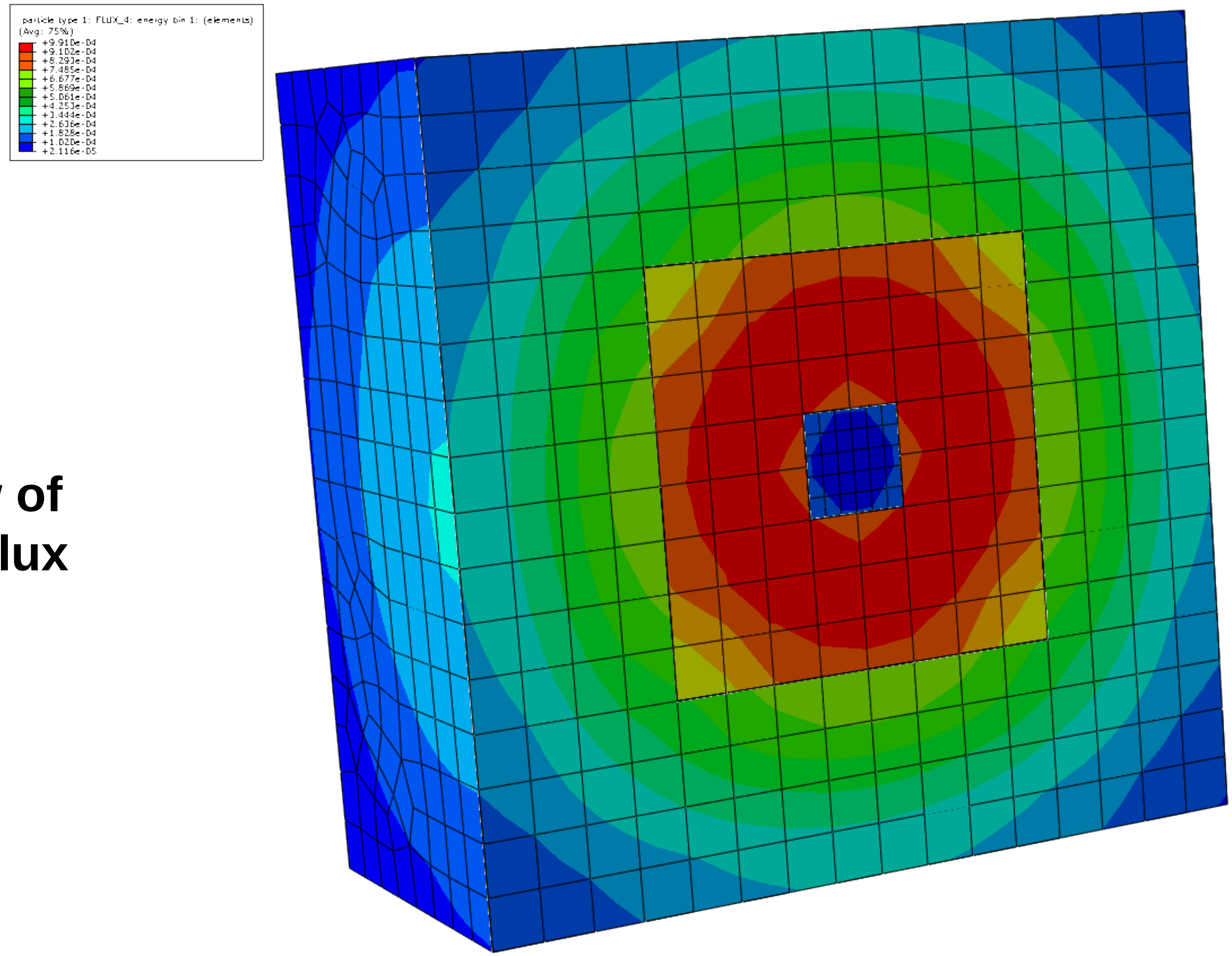
```
Concentric Cubes: 8x8x8 outer; 3 part, 1st order hexs
c
11  1  -2.03         0      u=2
12  2  -0.98         0      u=2
13  3   4.7984e-02   0      u=2
14  3   4.7984e-02   0      u=2
20  0                0      u=2
30  0              -99      fill=2
40  0               99

99  sph   0.  0. 0. 10.

c  Pseudo-concrete
m1   1001 -0.02  8016 -0.60  14000 -0.38
c
c  Water
m2   1001  2   8016  1
c
c HEU (Godiva) atom density:  4.7984e-02 at/b-cm
m3   92235  4.4994e-02  92238  2.4984e-03  92234
4.9184e-04
c
sdef pos= volumer
kcode 2000 1 10 20 90000
c
mode n
imp:n  1  1  1  1  1  1  0
c
```

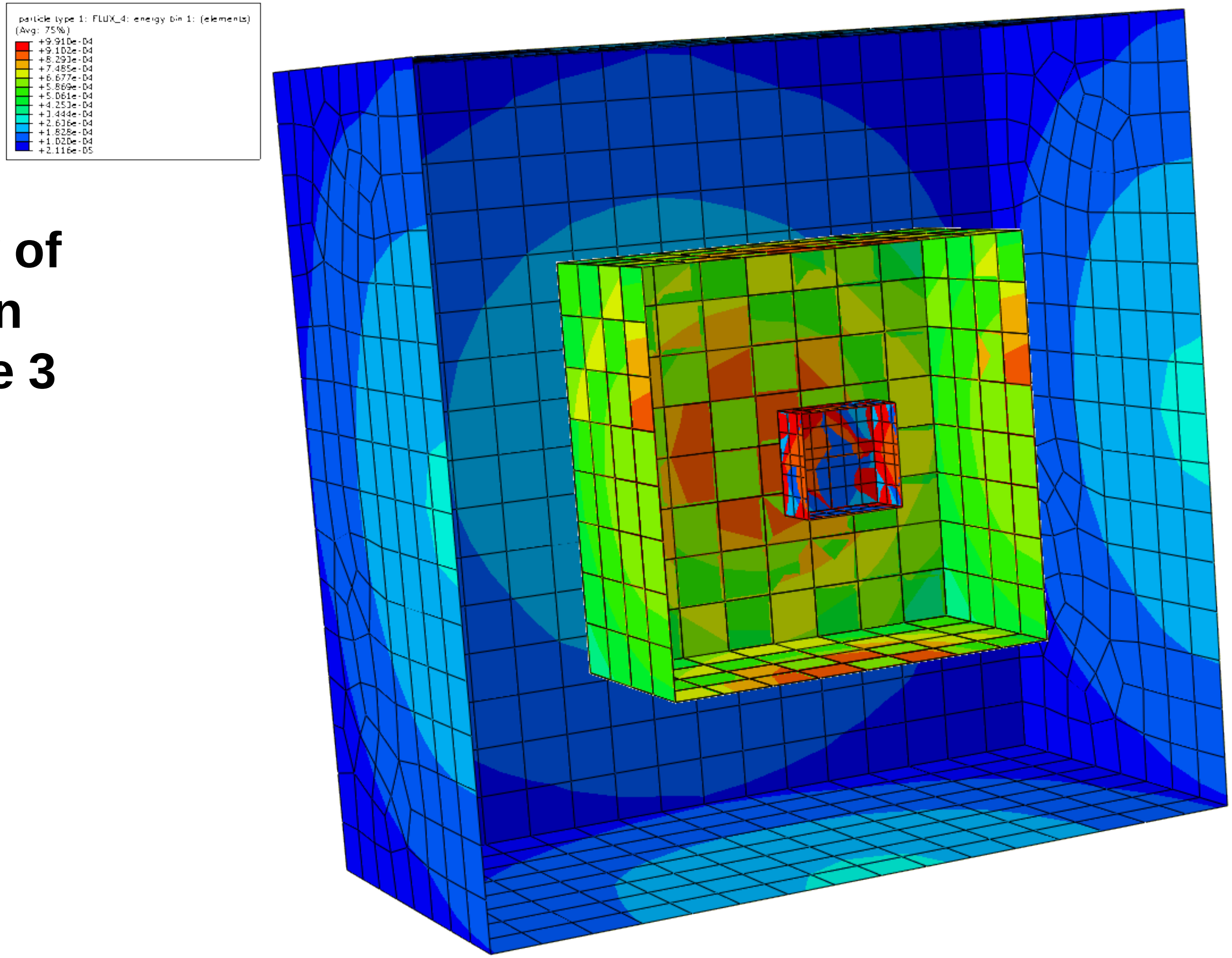
```
c Unstructured mesh data cards
c
embed2 meshgeo=abaqus
       mgeoin=um_8x8x8_3part_cube_s2s_hex_02.inp
       meeout=um_8x8x8_3part_cube_s2s_hex_02.eeout
       gmvfile=um_8x8x8_3part_cube_s2s_hex_02.gmv
       filetype=ascii
       background= 20
       matcell= 1 11  2 12  3 13  4 14
c
c
embee4:n  embed=1
embtb4  1e+39
embtm4  1.0
embeb4  1e+10
embem4  1.0
c
c Tallies
f14:n  11
```

Concentric Cubes: Abaqus/CAE Viz



Cut away view of total neutron flux

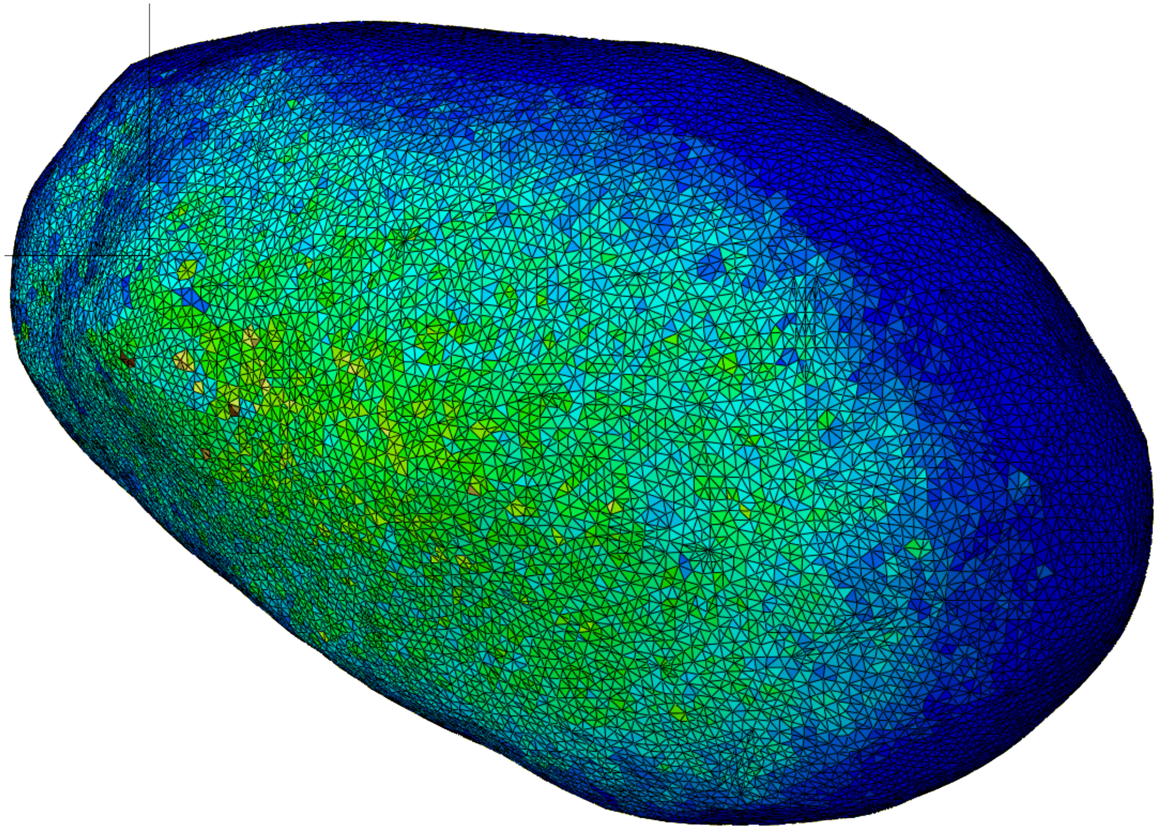
Concentric Cubes: Abaqus/CAE Viz



Cut away view of neutron flux on surfaces of the 3 parts

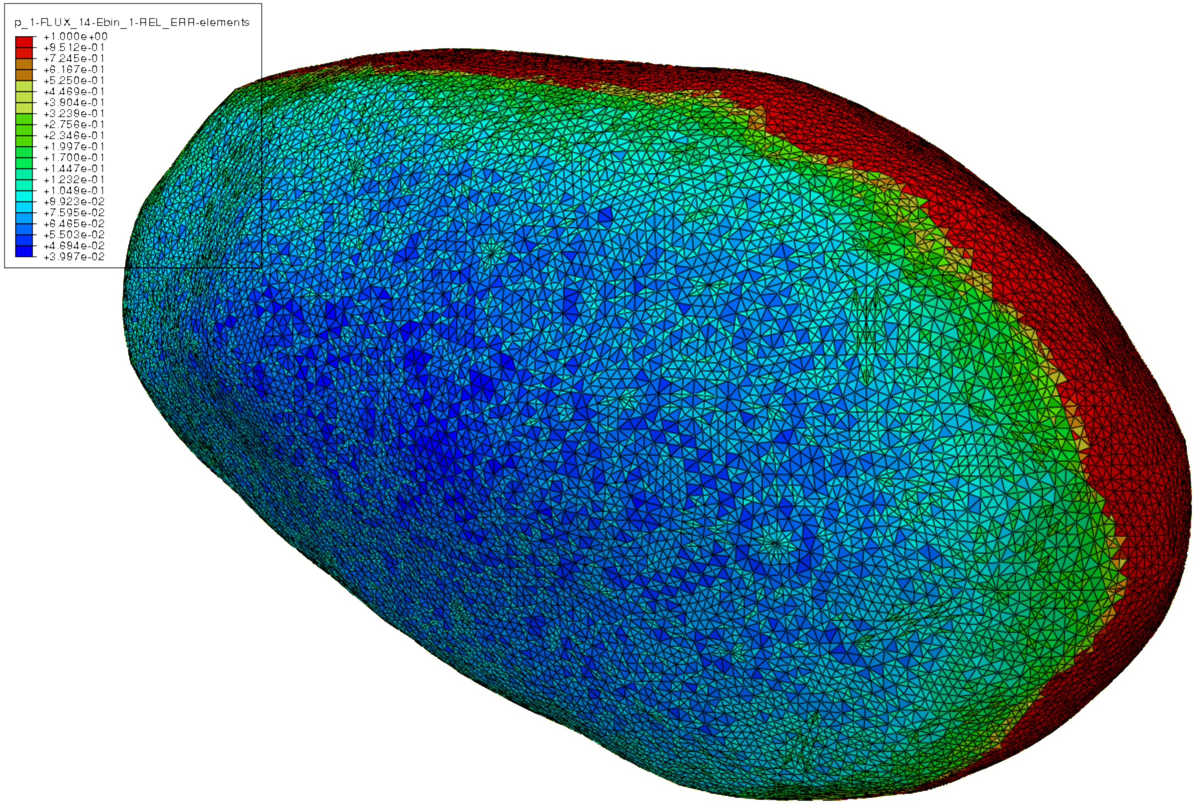
Itokawa Asteroid Ablation

- Fatman neutron and gamma time-dependent source located 650 meters from the asteroid's side.
- 128,886 1st order tets
- 1 billion histories
- Calculation time: 1 hour 50 minutes with 7 MPI slave nodes



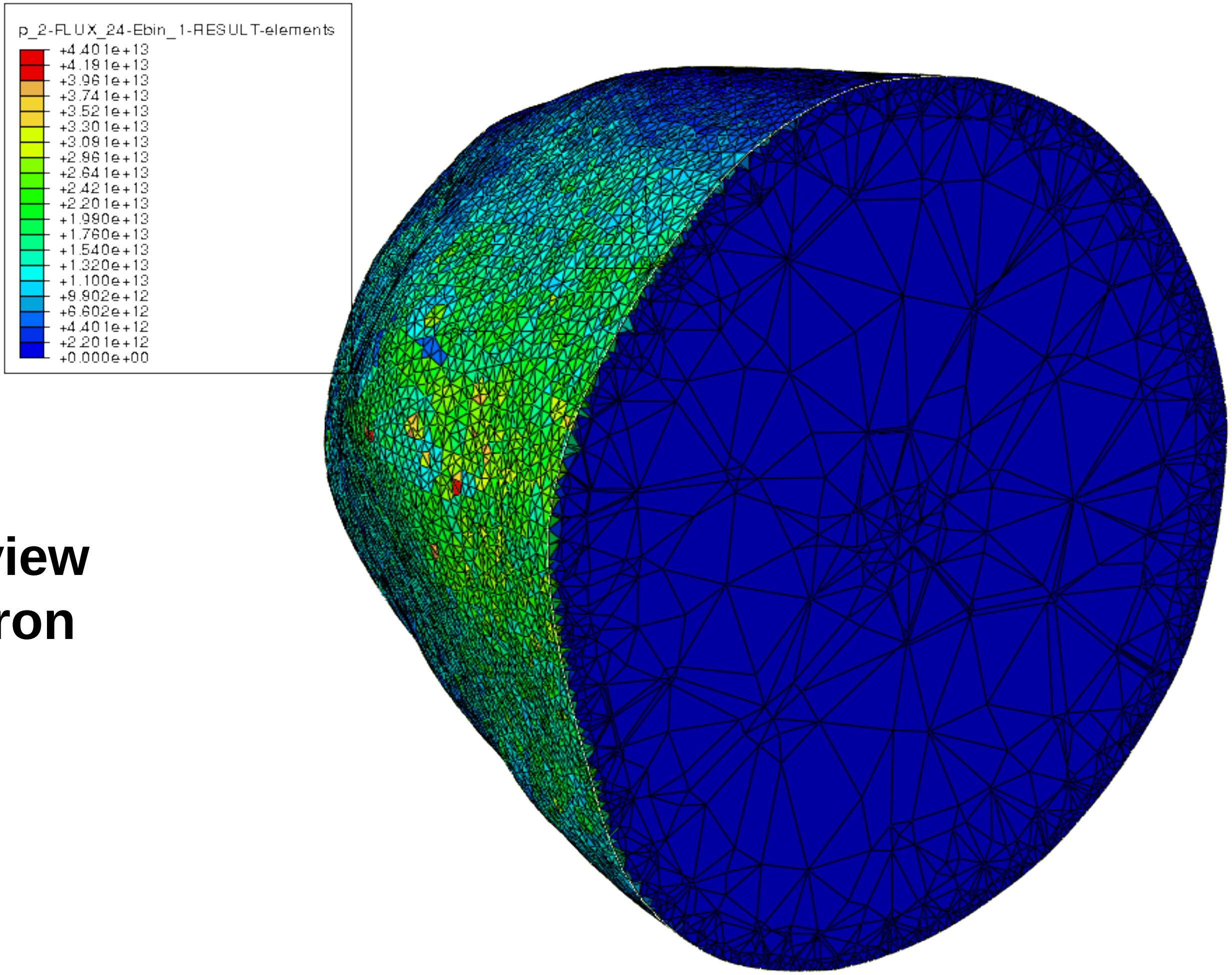
Total neutron flux

Itokawa Asteroid Ablation



Statistical error on total neutron flux in previous slide.

Itokawa Asteroid Ablation



Asteroid cutaway view showing total neutron flux.

ITER Demonstration Calculation

ITER model (20 degree section used for detailed analysis of diagnostic ports) calculation with MCNP6 Version 1.0

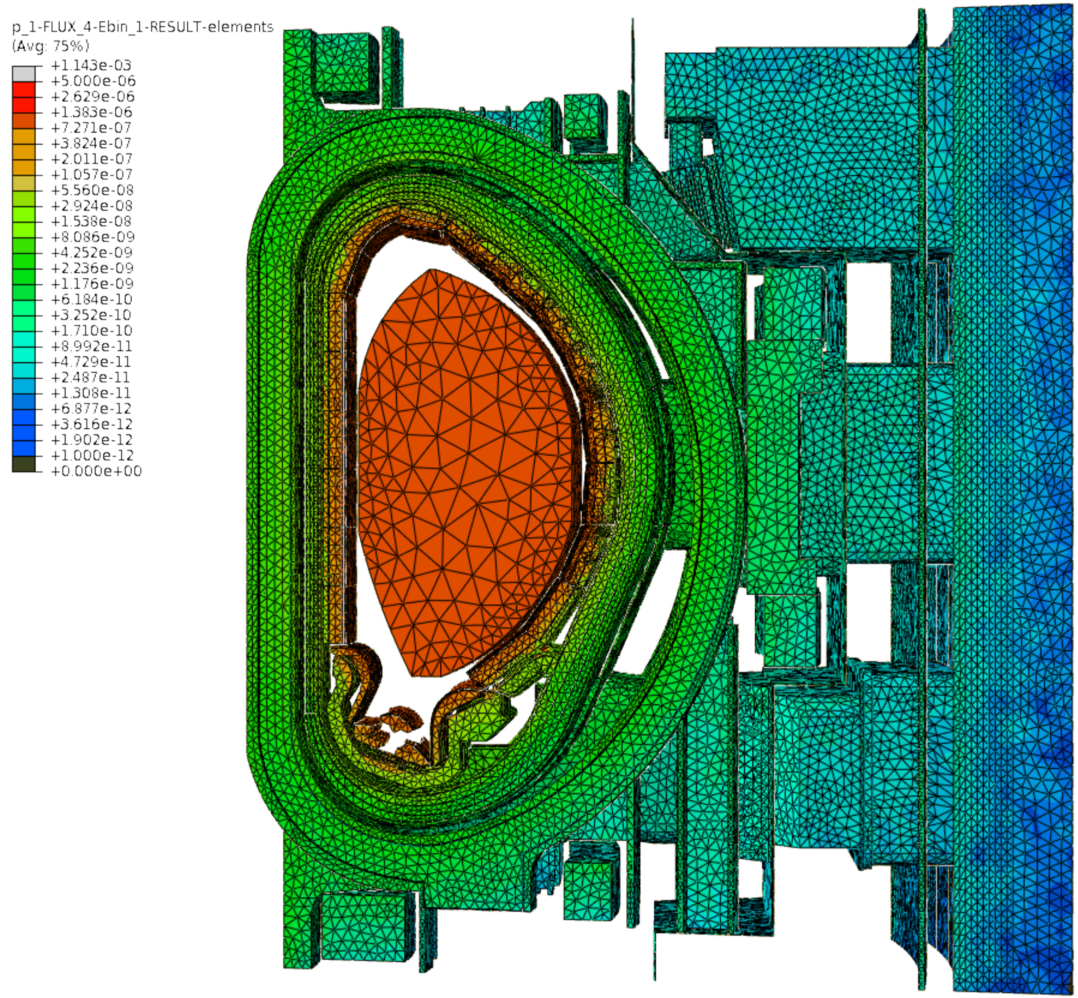
14.1 MeV mono-energetic neutron source using mesh volume source methodology.

Void region mesh removed to aid calculation performance and memory requirements (~4.5 GB/cpu).

2,073,968 1st order tets in 309 cells

Reflecting boundary conditions

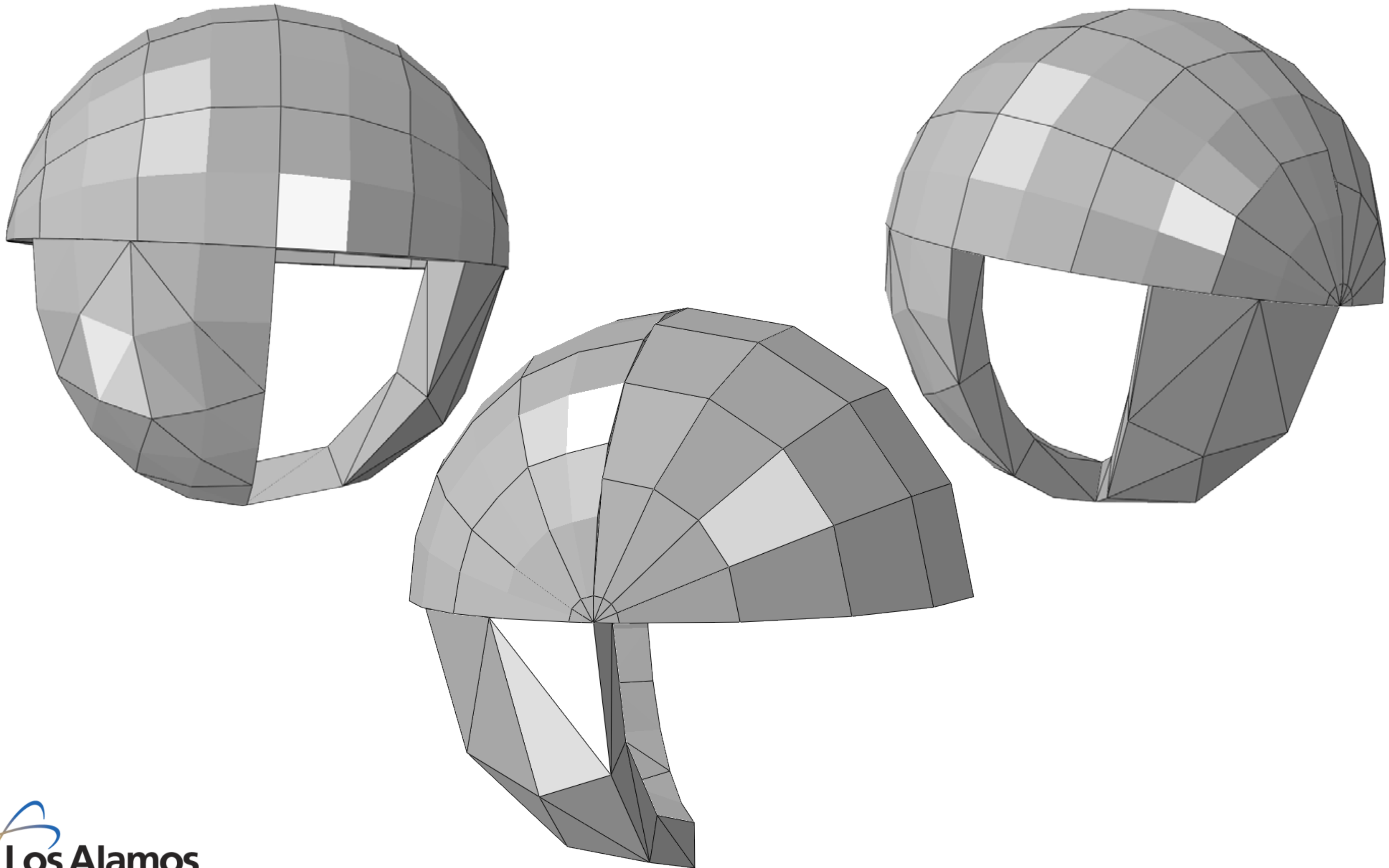
100 million histories run with 55 slave nodes. ~7.5 minutes setup time using parallel input processing.
~ 6.25 hours wall clock time with Intel Xeon E5-2670 chips @ 2.6 GHz running 64-bit Chaos Linux.



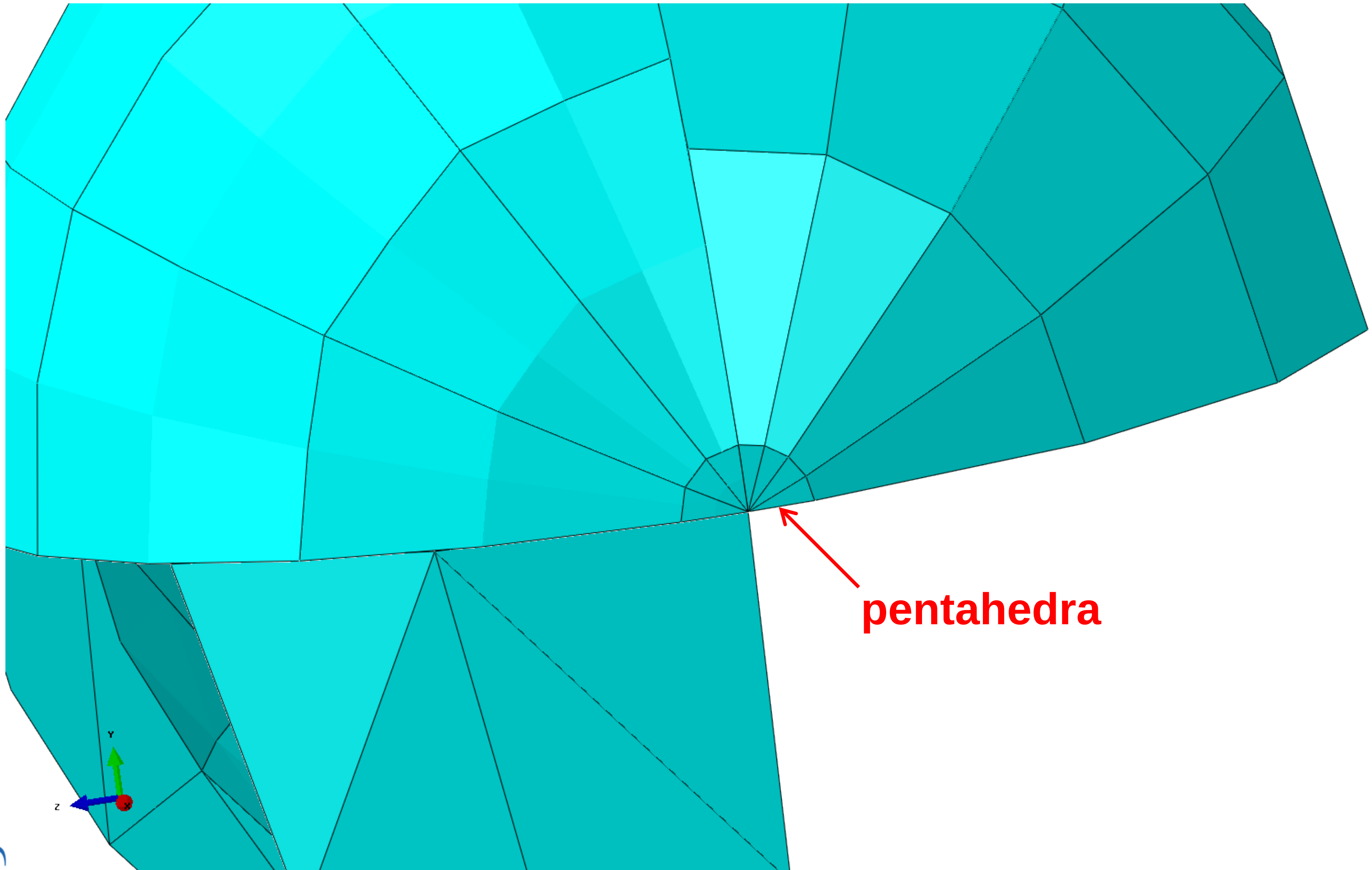
Total neutron flux

Extra Slides

Helmet: Mixed Element Types



Helmet: Mixed Element Types



MCNP User's Checklist For Abaqus Modeling

- **Define materials**
  - Appropriate name & number plus density
- **Define sections**
  - (primarily needed to get geometry vizualization correct in Abaqus/CAE)
- **Create simple parts**
  - Appropriate elsets: material, statistic, source
  - Section assignments
  - Mesh
- **Create any complex parts**
  - Inherits elsets from simple parts at time of creation
  - Mesh
- **Create assembly**
  - Translations & rotations
- **Write job file (i.e., the Abaqus .inp file)**
  - File name for input to MCNP
  - Job description for MCNP command line